Implementing data-driven maintenance at Stork IMM

Master's graduation thesis on increasing the reliability and productivity of Stork IMM injection molding machines by implementing data-driven maintenance techniques.

University of Twente

Faculty of Engineering Technology, Mechanical Engineering

Chair of Asset Management & Maintenance Engineering

Master thesis

Corné Lunshof

February, 2024

Project type:	Master thesis
Version:	Final version
Student:	Corné Lunshof
	s2641038
Company:	Stork IMM
Supervisors:	Hans van Vlierberghe
	Hans van den Broek
University:	University of Twente
Faculty:	Faculty of Engineering Technology
Specialization:	Maintenance engineering and organization
Research chair:	Asset management and maintenance engineering
Supervisor:	Jan-Jaap Moerman

Summary

Stork IMM, a manufacturer of plastic injection molding machines, wants to use data-driven maintenance to investigate the load on machines and to predict component failures. Current failures result in significant financial losses and downtime. Through the integration of data-driven maintenance, Stork IMM can minimize operational downtime and reduce warranty costs, thereby elevating client satisfaction levels and strengthening the organization's competitive edge. However, Stork IMM is struggling with implementing data-driven maintenance and is not alone; implementing data-driven maintenance is a known hurdle for small and medium-sized enterprises (SMEs). In this project, we implement data-driven maintenance in Stork IMM, focusing on the implementation process to learn how to improve it for SMEs. The study's central question is: How to implement and leverage data-driven maintenance in SME Stork IMM?

Other studies conducted with a traditional design research method fail at a certain phase, try overambitious changes in company processes, or remain with a few cases. Action research is applied as a research method to prevent these issues and gain unique insights. With action research, we can directly apply techniques or concepts, gather feedback, and improve. In the experimental part of the research, three data-driven maintenance use cases are implemented in separate research cycles.

During the implementation process, several hurdles for Stork IMM came to light. Bottlenecks were often related to the lack of system capabilities or skills required for data-driven applications, such as IT/OT convergence, organizational factors, and data completeness, consistency, and availability. Despite these hurdles, we implemented data-driven maintenance and were able to utilize the descriptive and diagnostic results valuably. Through data-driven analyses, we have diagnosed the degradation pattern of tie bars. The failure of the critical tie bar appears to be visible a million cycles before failure, and these three months provide enough time to deliver a new part. We have also reduced the load on specific frames so that the cracks do not tear further and the frames do not collapse until the new frames arrive. Finally, we demonstrated how productivity can be increased with downtime information.

The project uses technical frameworks demonstrating data-driven maintenance at different ambition levels. Implementation frameworks are used and evaluated, showing that they can be best improved by determining, in the preparatory phase, both the business needs and the final form that the maintenance technique addresses. Data-driven maintenance has been successfully applied by implementing the technology in a modular manner. By adopting a modular approach, we could progressively enhance the complexity of techniques and system capabilities. This allowed us to gradually develop the necessary skills and IT functionalities to achieve our required ambition level. The Agile and Modular Implementation Roadmap has been created to transfer this method and the coherence between different frameworks to future implementers. The roadmap combines the most important frameworks according to this research. The parallel visualization helps to understand the connection between technical and IT steps. Improvements to the frameworks and the modular and cyclical properties that ensure successful implementation have been incorporated into the roadmap.

Abbreviations

AI	Artificial Intelligence
APL	Rear clamping platen (achterplaat)
BI	Business Intelligence
CRISP-DM	Cross-Industry Standard Process for Data Mining
EMA	Exponential Moving Average
FEA	Finite Element Analysis
FMECA	Failure Mode Effect and Criticality Analysis
FTA	Fault Tree Analysis
IMM	Injection Molding Machine
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Indicator
LSP	Moveable clamping platen (losse spanplaat)
MES	Manufacturing Execution System
MTBF	Mean Time Between Failure
NLP	Natural Language Processing
OEE	Overall Equipment Effectiveness
OEM	Original Equipment Manufacturer
OT	Operation Technology
PDM	Product Data Management
RUL	Remaining Useful Lifetime
TTF	Time to Failure
VSP	Fixed clamping platen (vaste spanplaat)

Contents

1. Introduction

1.1. Initiation of the assignment

Production companies rely on their injection molding machines. One case in this study concerns a machine that makes ice trays at a rapid pace 24 hours a day to meet the demand of the upcoming summer but is in danger of collapsing due to cracks in the frames. The machine is on a different continent than Stork IMM and new frames have a lead and transport time of almost a year. Data-driven maintenance must help condense the frames up to replacement. The failure of this asset carries significant financial consequences, highlighting the importance of data-driven maintenance to meet today's high reliability standards (Tiddens, 2018). Any time a machine experiences a breakdown, it causes frustration for the manufacturer and poses a risk of generating negative perceptions about the brand in the competitive market. It's crucial to prevent such occurrences to maintain a positive reputation and standing in the industry.

Stork IMM and its clients could have already realized substantial cost savings by implementing data-driven maintenance, like the production, transport, and machine overhaul as in the previous paragraph. In addition, expensive rapid air transport of an 8-meter-long, 2000-kilogram tie bar could have been avoided if the condition of these tie bars without a spare part had been known. Preventable costs underscore the pressing need to get up to speed as quickly as possible. Data-driven diagnoses and forecasts result in cost savings and increased customer satisfaction through less downtime.

Data-driven and predictive maintenance are much-discussed topics in industry and academia. Advancements in technology, especially in the world of the Internet of Things (IoT) and Artificial Intelligence (AI), have made predictive maintenance very relevant nowadays (Sensorfy B.V., 2023). Because we see new IT solutions all around us, the potential of these IT techniques is continuously demonstrated, which leads to new projects. In this project, we focus on data-driven maintenance. Compared to predictive maintenance, the goal of data-driven maintenance is not only to predict and optimize the maintenance time, but also to improve asset performance and provide diagnoses. However, this is not yet being utilized because the data-driven facilities still need to be developed.

1.2. Stork IMM

Stork IMM (Injection Molding Machines) manufactures high-speed injection molding machines, mainly for thin-walled plastic products. There are several manufacturers of injection molding machines in Europe. There are competitors of Stork IMM in Europe with significantly larger sales. Compared to these competitors, Stork IMM distinguishes itself in high quality, flexibility, and the possibility of specific customer options. There are also other European competitors with comparable sales. To these customers, Stork wants to distinguish itself in high quality, production speed, and reliability.

All office and manufacturing operations are based at the Stork IMM headquarters in Hengelo. Stork IMM's core activities are selling new machines, providing service for machines in usage, and overhauling used machines. The engineering department includes order work, product management, and research and development. The service department provides service worldwide, with a significant part in Europe.

An organizational chart is shown in Figure 1 to illustrate the roles of the involved departments. The use cases in this project are quality problems for which a structural solution is developed. For this reason, the assignment is carried out in the engineering department, not the service department. In this project, the mechanical and software sub-departments of engineering are mainly involved. The graduation student is responsible for developing the technical content and algorithms, while the software department handles the integration of logging software and IoT systems into the machines and servers. The designated customer relationship sales employee handles customer meetings, contacts, and proposals.

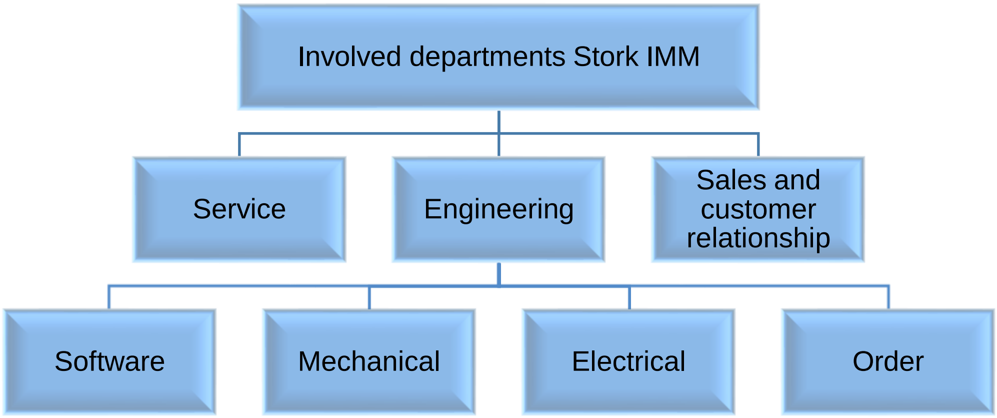


Figure 1, Involved organizational departments

Stork IMM has about 110 employees, making it a medium-sized company (European Commission, 2023). These employees include every position, among engineering, procurement, sales, service, finance, revision, machining, assembly, and service technicians. Because of Stork IMM's extensive technical know-how and strong collaboration with clients, the company also contributes significantly to various developments. It fits in with the Dutch highly technical culture, in which Stork IMM, a small company, takes a leadership role in pioneering new functions and high-tech developments.

1.3. Practical and theoretical relevance

Stork IMM’s machines produce 24 hours per day plastic parts in production plants. Extraordinary high loads on components all over the machine cause all kinds of defects. The machine has standard wear parts, parts with fatigue defects, and special events such as blasts or abuse that lead to overload. The different types of defects occur regularly in unknown patterns. For manufacturers, unplanned downtime is costly due to material costs, service costs, and especially missed production time. Due to the wide variety of parts that can fail, it is unrealistic for production companies to have all spare parts in stock. Even Stork IMM, as OEM (original equipment manufacturer), does not have all potential spare parts in stock. As a result, reactive maintenance combined with long lead times can result in extended downtime. Long downtime is costly for manufacturing companies and leads to lower customer satisfaction. Monitoring reliability can ensure higher customer satisfaction through uptime, proactive service, and lower maintenance costs. Moreover, reliability is one of the most crucial performance indicators of injection molding machines. Therefore, Stork IMM can distinguish itself from its competition by offering data-driven maintenance.

IT solutions such as IoT, big data, AI, and machine learning are game changers for data-driven maintenance, making it a very active research topic (Krishna Durbhaka & Selvaraj, 2021). However, the new, technically complex systems make implementation difficult for SMEs (Mainnovation, 2018; Matt et al., 2020; Van Eijk, 2023). Implementing data-driven maintenance is difficult for SMEs (Small and Medium-sized Enterprises) because of extra challenges in the company's IT infrastructure, limited resources such as personnel and budget, and a lack of experience with the complex technology (Matt et al., 2020). The company's uncertainty regarding the initial steps and the strategy for effective implementation of data-driven maintenance initiates this project. Researching implementing data-driven maintenance in SMEs and making it more tangible can greatly contribute to bridging the gap between theoretical knowledge and practical implementation.

The costs and benefits of data-driven maintenance are often not explicitly defined or evaluated, making it hard to define a solid business case (Tiddens et al., 2015). Current literature hardly highlights interim results and possibilities with the data received from monitoring processes. These interim and side results can be essential for SMEs that want to get as much value as possible from the required investment. Better reflection on the implementation and interim results is practically and theoretically important for the future of data-driven maintenance in SMEs.

1.4. Research questions

The previous part of the introduction discussed the initiation, background, and relevance of the problem. Research objectives and questions can be formulated based on this problem.

Research objective

The objective of the research is to implement data-driven maintenance in Stork IMM. Several cases in the injection molding machine have great potential for data-driven maintenance. Both Stork IMM and its customers can benefit from the technique. However, data-driven maintenance has not yet been implemented due to various obstacles, a general problem for SMEs. There is an academic interest in improving the implementation of data-driven maintenance in SMEs, thus narrowing the gap between theory and practice. SMEs struggle with complex IT systems and require skills in different aspects. The literature review examines all factors surrounding this problem to reinforce the research gap. In summary, the research goal is to implement and leverage data-driven maintenance in Stork IMM machines and improve the accessibility of data-driven maintenance for SMEs.

Research questions

Stork IMM faces challenges implementing data-driven applications and data-driven maintenance on injection molding machines. Using this valuable opportunity of applying data-driven maintenance in Stork IMM and learning from the implementation process, the research question with sub-questions are:

How to implement and leverage data-driven maintenance in SME Stork IMM?

1. How can data-driven analysis techniques be used to monitor the condition of Stork IMM injection molding machine components?
2. How can data-driven maintenance be used to increase the reliability of Stork IMM machines?
3. What organizational needs and requirements should be considered to implement data-driven maintenance effectively in Stork IMM's specific operational environment?

1.5. Thesis outline

The research starts by establishing a theoretical framework around the subject. This literature review discusses condition-based maintenance, predictive maintenance, and implementing data-driven maintenance. The literature review will already give a good impression of the first two research questions. Subsequently, the core of the research is shaped by defining the research methodology. Chapter 4 discusses the preliminary steps for the data-driven maintenance use cases. In this chapter the injection molding machine is analyzed and the IT system is designed. Chapter 5 describes use cases where data-driven maintenance is implemented. These use cases are the core of the research and should provide valuable insights into data-driven maintenance and the implementation process in Stork IMM. We assess the tools outlined in the most recent literature and identify shortcomings of Stork IMM that arise during implementation. Experiencing this implementation is the input for answering the last research questions. This is evaluated and interpreted in the discussion in Chapter 6. With the findings interpreted in the discussion, the Agile and Modular Implementation Framework is presented, with the best frameworks and methods according to this research that are useful for future implementers.

2. Literature review

The literature review serves as a theoretical framework for the current state of knowledge about maintenance strategies, predictive maintenance, and the implementation of data-driven maintenance strategies.

2.1. From data to condition-based maintenance

Condition-based maintenance is a maintenance policy. This paragraph explains the development of condition-based maintenance, why it has become interesting, and its relation to Industry 4.0.

Condition-based maintenance

Maintenance is often crucial in asset management. As described by NEN-EN 13306:2019, maintenance is *a combination of all technical, administrative, and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function* (NEN-EN, 2019). In injection molding machines, losing precision due to a worn part can be approached as a function loss, initiating maintenance. Maintenance policies can be categorized into two strategies: corrective and preventive maintenance (Tinga, 2010). Corrective maintenance, also frequently referred to as reactive maintenance, is commonly dismissed as an undesirable option that leads to significant downtime and costs. However, this perception is somewhat unjustified, as it can be the most cost-effective strategy in many cases. If monitoring is infeasible or spare parts have relatively low costs, reactive maintenance is the most suitable strategy.

Preventive maintenance replaces parts before they cause unplanned downtime. Preventive maintenance increases the reliability of assets and the operation level of firms (Tiddens, 2018). However, when using time or usage-based preventive maintenance schedules, components are only partially utilized, downtime for replacement is too long, and too many labor hours are spent (Tinga, 2010). Condition-based maintenance is a more proactive preventive maintenance policy. The preventive interval is optimized by considering the condition of the component. There are several ways to determine the condition. Ideally, the condition of the part is measured directly. However, in practice, measuring the condition is often much more complicated than, for example, a load. Figure 2 shows approaches to iterate the condition.

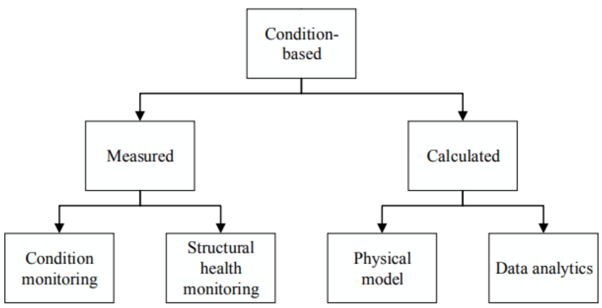


Figure 2, Category distinction of maintenance strategies (Tiddens, 2018)

The condition of parts can be determined differently, as shown in Figure 2. The chosen strategy depends on the technical knowledge about the failure mode, the feasibility of measuring the condition, and data availability (Tiddens, 2018). Methods to assess the condition of components are discussed in more detail later in this chapter. More maintenance strategies exist than are presented in this literature review. For example, opportunistic maintenance occurs regularly in practice, where the machine's downtime is used to replace other parts. A typical opportunistic maintenance example is that the car's water pump is often replaced when the timing belt has reached its defined useful life.

The role of industry 4.0

Industry 4.0 is a collective term of technological concepts for smart solutions, such as Cyber-Physical Systems, IoT, Data Mining, and Big Data (Wang, 2016). These digital techniques combine sensors with systems that review and determine the condition. So Industry 4.0 enables IT-based communication between machines and services (Wang, 2016). Smart products or systems have monitoring, optimizing, and self-diagnosing capabilities to enhance the development of condition-based maintenance (Tiddens, 2018). An IoT infrastructure facilitates data science appliances. A typical data science process for predictive maintenance is shown in Figure 3.

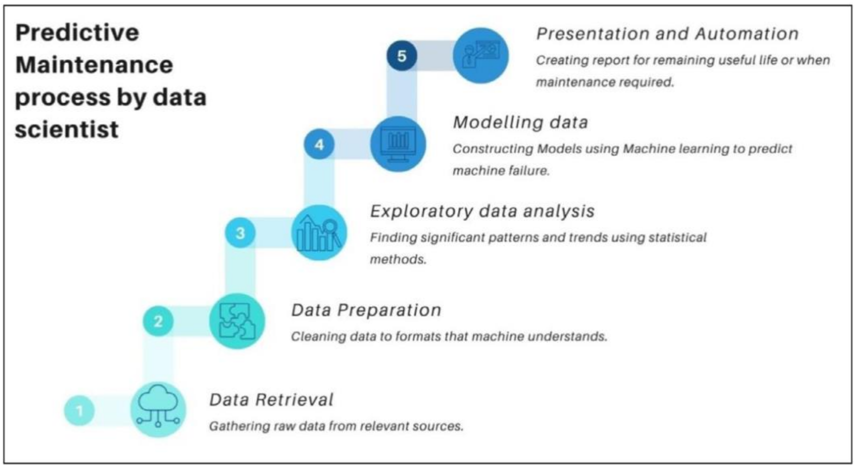


Figure 3, A typical data science approach (Sajid et al., 2021)

CRISP-DM stands for Cross-Industry Standard Process for Data Mining and is a widely used and successful data mining model. The CRISP-DM model helps practitioners across the industry to make complex data mining projects successful and effective (Chapman et al., 2000). The model is shown in Figure 4 and is self-explanatory for people with some data science knowledge.

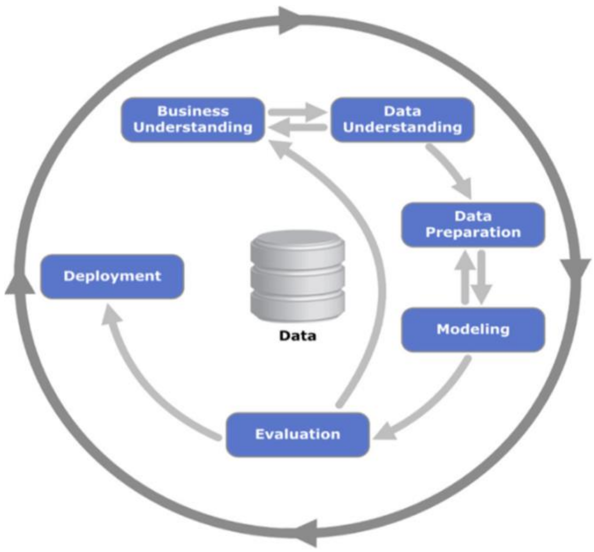
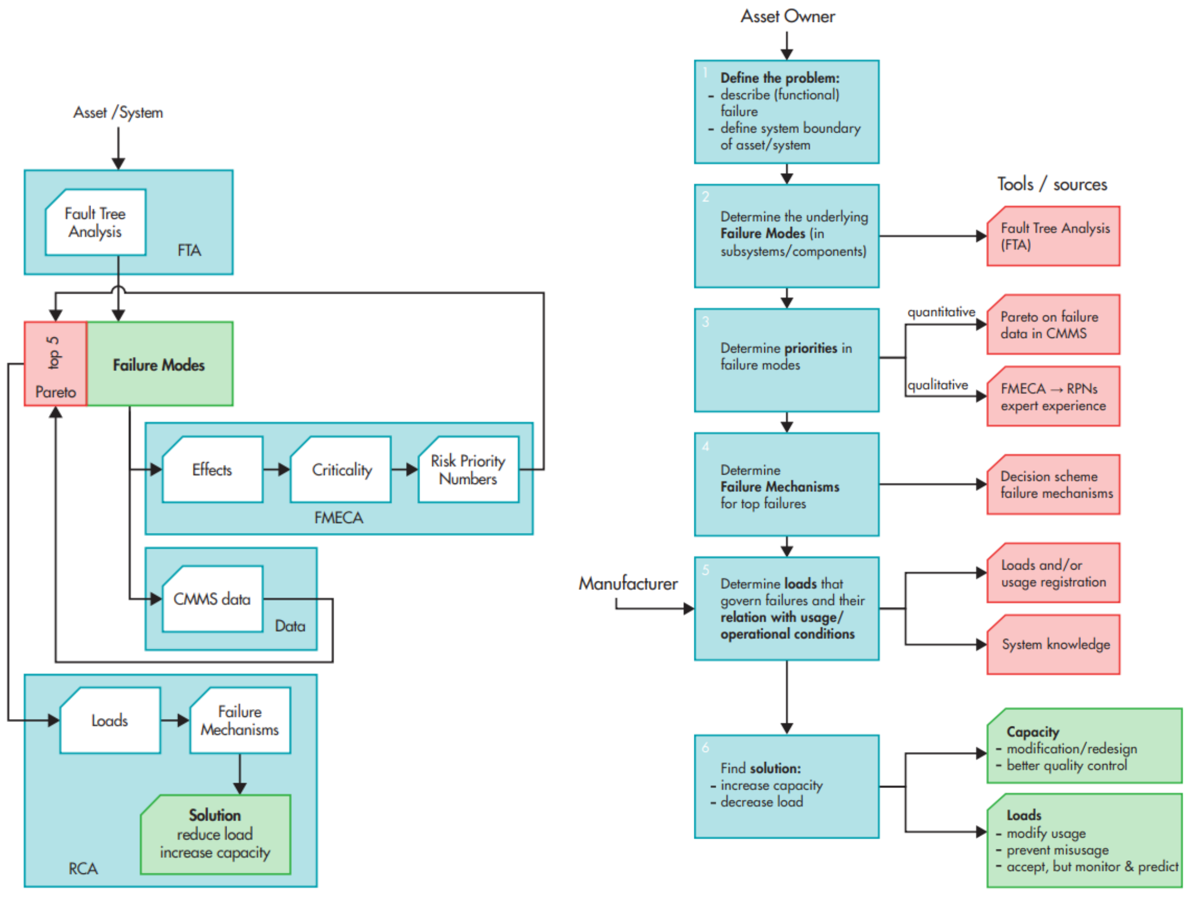


Figure 4, CRISP-DM (Cross-industry standard process for data mining) (Chapman et al., 2000)

Selecting suitable components

The best-known methodologies to identify critical components and failure modes in a system are a fault tree analysis (FTA) and a failure mode, effect, and criticality analysis (FMECA) (Tinga, 2012). An FTA is a top-down approach, where a system is divided into smaller sub-levels to identify possible faults. An FMECA is a bottom-up approach where separate failure modes are more thoroughly analyzed. The role of these methods is shown in Figure 5. In the process guidelines, the incoming pointed arrow for manufacturers is an approach where the manufacturer, in this case Stork IMM, can make a difference.



(a) (b)

Figure 5, (a) failure analysis process, and (b) process guidelines (Tinga, 2012)

A funnel approach is proposed by (Tiddens, 2018) to select suitable candidates for data-driven maintenance. The funnel process has three steps: criticality classification, showstopper identification, and focused feasibility. The criticality classification filters components with a low frequency of failure and significant consequences (Tiddens, 2018). In the second step, showstoppers are identified by assessing the technical, economic, and organizational feasibility of the case. The last step, a focused feasibility study, investigates the data-driven maintenance processes in detail for the cases. The specific steps are described in the following chapter.

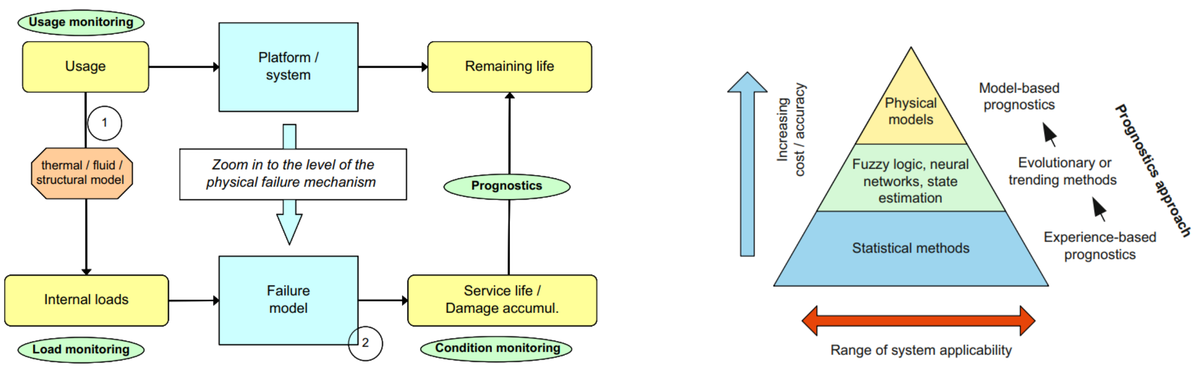
2.2. Predictive maintenance

Predictive maintenance, as the name says, predicts maintenance. A key distinction from condition-based maintenance is that maintenance actions are predicted rather than looking at the asset condition. Predictive maintenance is defined by NEN-EN 13306:2019 as *condition-based maintenance carried out following a forecast derived from repeated analysis or known characteristics and evaluation of the significant parameters of the degradation of the item* (NEN-EN, 2019). This paragraph compares different predictive maintenance frameworks.

The core of predictive maintenance

Predictive maintenance is an advanced maintenance strategy where data-driven analytics are used to maximize the service life of equipment (Wang, 2016). A forecast or decision is concluded from the measured conditions or loads. Typical concepts that are an outcome of such a forecast in predictive maintenance are remaining useful life (RUL), time to failure (TTF), and mean time between failure (MTBF). The result often has a particular uncertainty because these are predicted values. As defined by NEN-EN 13306, the prediction is based on a statistical failure distribution or a physical degradation model (NEN-EN, 2019).

Asset data is vital to perform either statistical or degradation predictive prognostics. Figure 6 illustrates progressive steps in increasing accuracy to monitor the condition from usage to condition of components. The usage-to-load and load-to-life relation are noted by numbers 1 and 2, respectively. These relations hold great importance, as they can only be established if the physical background of the loads and failure mechanisms are understood (Tinga, 2010).



(a) (b)

Figure 6, (a) iteration to remaining life, (b) types of prognostic approaches (Tinga, 2010)

Directly assessing the condition of components minimizes remaining uncertainty. However, gathering data in earlier stages necessitates more models, potentially reducing accuracy (Tinga, 2010).

Data-driven maintenance frameworks

An overview of six different data-driven maintenance frameworks is made by Van Eijk (2023) and shown in Figure 7. All frameworks are approximately similar in the sequential phases: critical failure mode selection, identification of degradable mechanisms, data acquisition, data analysis, evaluation, and decision-making. Perhaps the prognostic activity was better placed in the evaluation phase, under the interpretation of evaluation on the analyzed data. Even though there are similarities in the individual action points during each phase, differences are also visible. Each framework has a different focus or uses other techniques. In the coming sections, two frameworks are examined. The methods and techniques in these frameworks cover most of the ideas of all frameworks.

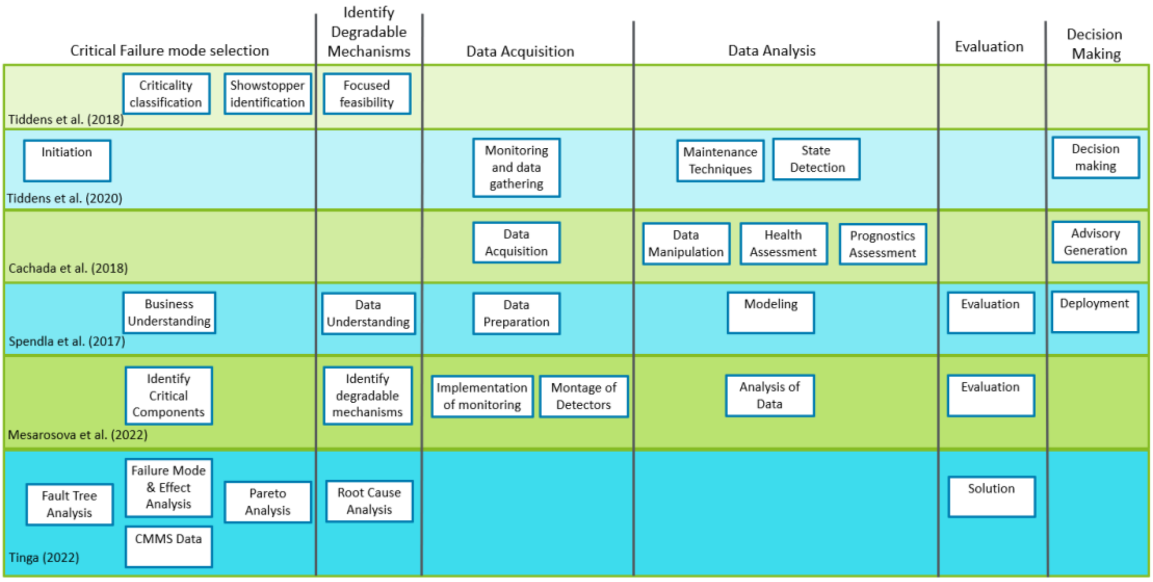


Figure 7, Data-driven maintenance frameworks (Van Eijk, 2023), frameworks from (Cachada et al., 2018; Mesarosova et al., 2022; Spendla et al., 2017; Tiddens, 2018; Tiddens et al., 2020; Tinga, 2012).

Primavera predictive maintenance framework

A generic predictive maintenance process model is introduced by the research group PrimaVera, shown in Figure 8. PrimaVera's goal in the presented model is to provide a generic, effective, and efficient framework that supports asset management by predictive maintenance (Ton et al., 2020). After the figure, we discuss the actions and steps in more detail.

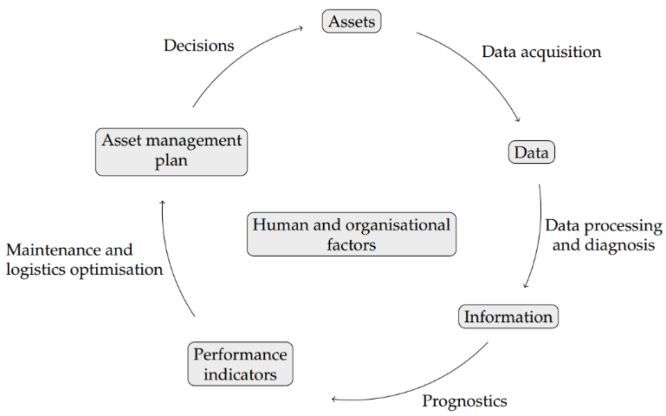


Figure 8, PimaVera generic predictive maintenance process model (Ton et al., 2020)

Incorporating data-driven workflows in companies is difficult (Tiddens et al., 2015). For each specific case, companies struggle to determine the sensors, sensing strategy, the required amount of data, data accuracy, and data type (Tiddens et al., 2015). In addition, advanced IT techniques such as AI and machine learning require large amounts of high-quality data. However, in reality, data tends to be messy, incomplete, unaligned, inaccessible, and inconsistent. (Ton et al., 2020). The ownership and structure of data are often missing (Ton et al., 2020).

Acquired data from sensors need processing to make the data useful. Data cleaning, transforming, and feature extraction are required to acquire information from the data. Data science uses data mining to turn raw data into useful information (Sajid et al., 2021). This data process is necessary to perform regression-based techniques and statistical calculations. Figure 3 shows a typical data processing structure for data-driven maintenance.

In the prognostics phase, algorithms are created to predict future failures. This step involves measuring relevant performance indicators, such as remaining useful life and reliability (Ton et al., 2020). A statistical approach to predictive maintenance needs an extensive and complete set of information, including data sets where failure arises. From unusual failures, obtaining varying sets of failures that form a basis for machine learning identifications can cost much time. As a result, the statistical approach is often combined with domain knowledge and physics-based prognostics (Tinga, 2010). An estimated lifetime or reliability is the conclusion from predictive prognostics to optimize the maintenance moment.

Data-driven maintenance techniques framework

The maintenance techniques framework from Tiddens (2018) is presented in Figure 9. The four required elements for data-driven maintenance decision-making are indicated by letters A through D. Progressive steps in accuracy and complexity in steps B, C, and D from left to right have the same ideology as the steps in Figure 6. A technology push or a decision pull initiates the start of a project. In a decision pull, data-driven maintenance is used to achieve a requested goal. In a technology push, data-driven maintenance technology is used to demonstrate a use case. The second phase involves monitoring and collecting data.

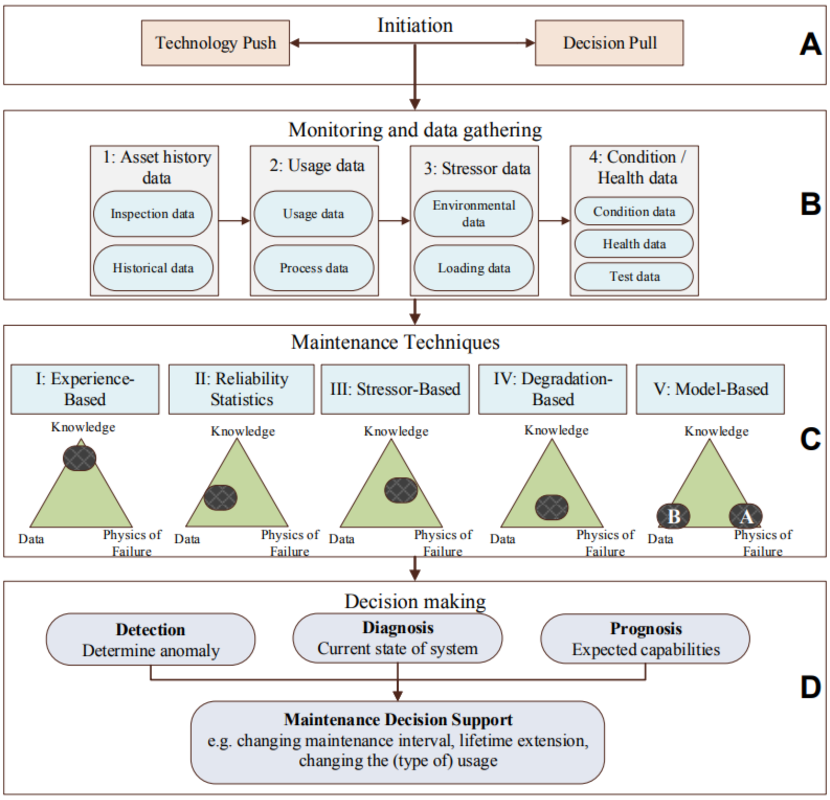


Figure 9, Maintenance techniques framework (Tiddens, 2018)

The choice of maintenance technique depends on the knowledge about the failure modes and the data gathering method from step B. Phase C is comparable to the information and prognostics phases in PrimaVera's framework in Figure 8. The maintenance techniques are explained with the same list signs as used in Figure 9 in the following summary:

I. Experience-based maintenance depends on historical knowledge of the functioning of the machine. This technique is commonly used in Stork IMM. Experienced staff know from experience which solutions do and do not work in situations with a specific context. They assess the loads, variables, and the average performance of parts.
II. Referring to the definition of predictive maintenance, a prediction can be derived from repeated analysis, forming reliability statistics (NEN-EN, 2019).
III. Stressor-driven predictions utilize historical data with stressor information, including factors like temperature, force, or speed. These supplementary elements consider diversities in the environment and operations, resulting in estimations concerning the projected durability of an average system in a specific context (Tiddens, 2018).
IV. Degradation-based maintenance combines measured data with the physics of a failure (Tiddens, 2018). For example, measuring the wear depth of a brake pad can form a degradation pattern, from which the remaining lifetime can be extrapolated.
V. Model-based predictions embrace two types of approaches that can be employed to estimate the expected remaining lifetime of a specific system under specified conditions (Tiddens, 2018):
    a. In a physics-based model, the prognostic parameter is determined by employing a physical model of the degradation mechanism that utilizes direct sensing of the loads or usage governing the critical failure mechanisms of individual components (Tiddens et al., 2023).
    b. In a data-based model, measured data is used to update the degradation pattern depending on the measured load or condition. Algorithms aim to derive patterns, relations, and anomalies by comparing them to historical data (Tiddens, 2018).

The major difference in the models presented is how a prognosis is made. The technique's complexity and accuracy increase with each step of the above-presented list. For example, steps 4 and 5 differ because step 4 only uses a degradation model, but step 5 also includes boundary conditions to adjust the degradation rate. Selecting distinct techniques relies on the feasibility of acquiring measurements related to the condition or load and the extent of available knowledge about historical or statistical data. The outcome of the models can support the last step in deciding to carry out the maintenance. The maintenance models are mapped in Figure 10 against the required data types and resulting levels.

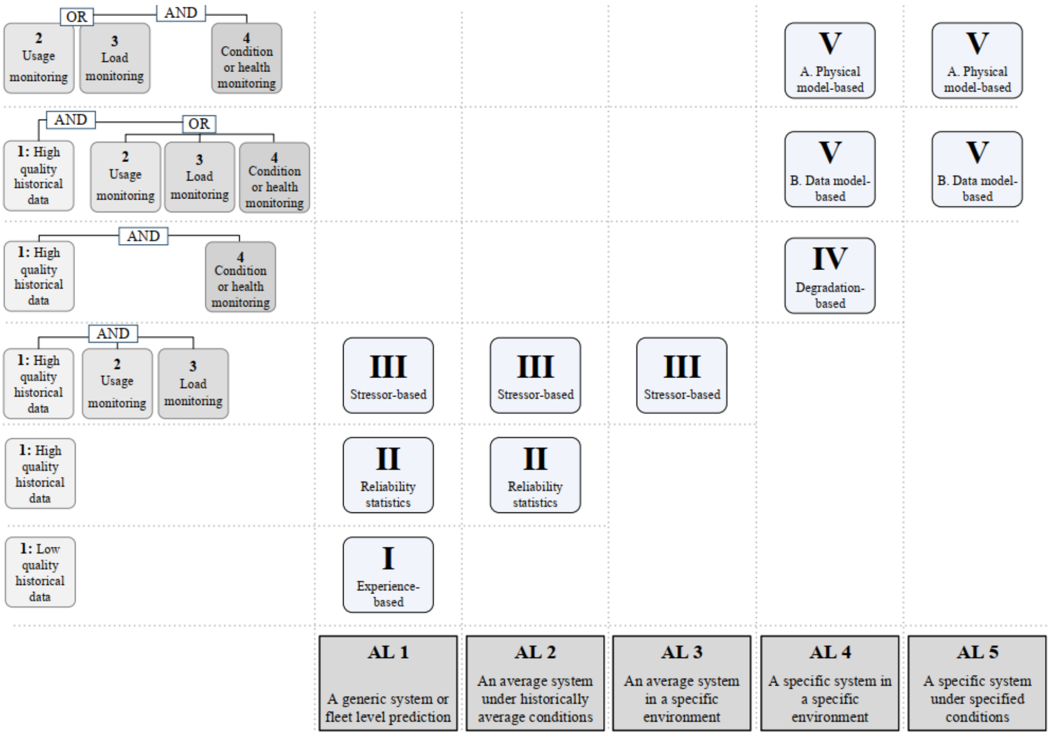


Figure 10, Mapped maintenance approaches to ambition levels and data types (Tiddens, 2018)

Data-driven maintenance for OEMs

Adding data-driven maintenance techniques prevents maintenance, while maintenance is an important source of income for Stork IMM. It also makes the machine more expensive in the already highly competitive market. Machine downtime is a problem for manufacturers, not directly for the OEMs. The reasons why OEMs offer data-driven maintenance can vary. Examples found in the literature (Ingemarsdotter et al., 2021; Sensorfy B.V., 2023):

- Increase service to customers from a distance
- Competitive advantage for OEMs
- Make machine ready for data-based appliances
- Increase customer satisfaction by increasing uptime
- Warranty contract

The form depends greatly on how the technical application is sold. A well-known example from the aircraft industry is using engines paid per hour, thus changing the product from purchase to service. Use cases performed by (Ingemarsdotter et al., 2021) show companies that sell uptime contracts, cloud applications for seeing analyses, and services based on connectivity. The form to sell is market and asset specific.

Predicting

In theoretical literature studies, RUL prognostics are made using AI algoritmes that use deep Gaussian processes and neural networks with Monte Carlo dropout (Mitici et al., 2023). These approaches are not yet widely seen in practical applications and because of the significant statistical dependence, lots of learning data is required. Statistical probability distributions mostly seen in literature to describe a degradation pattern are a generic gamma process, a Wiener process, or a non-homogenious Poisson process (Mitici et al., 2023). An example of RUL prognostics using a probability distribution is shown in Figure 11, where a degradation threshold is compared to the usage. By extrapolating the degradation pattern, a probability of failure of the component over time can be calculated.

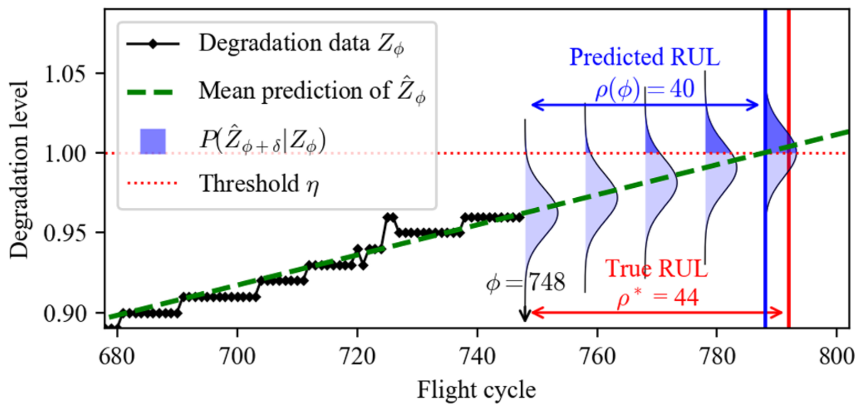


Figure 11, RUL prognostics example (Lee et al., 2022)

2.3. Implementing data-driven maintenance

In the previous section, we saw what data-driven and predictive maintenance entails. In this section, we look at the skills required to implement data-driven maintenance successfully.

Maturity models

Maturity models from academia and industry (Mainnovation, 2018; Van de Kerkhof, 2020), shown in Appendix B, identified vital company capabilities for data-driven maintenance. Vital capabilities are technological, IT, and organizational factors. The maturity is assessed from no maturity to world-class maturity on specific sub-level capabilities. Using the maturity models, companies can relatively quickly know where to invest for the next step to get closer to data-driven maintenance. The models are based on the idea that all defined criteria must be around at least a certain maturity to achieve a certain level. However, maturity models must be assessed by staff with knowledge of the subject. This knowledge is also required to perform the connection between the points. The maturity models indicate points to focus on, not specific tasks or instructions to gain maturity.

Maintenance techniques

As presented in section C of Figure 9, several maintenance techniques require different types of data monitoring, knowledge of the asset, and technological skills. The selection of the maintenance technique is primarily influenced by a feasible data monitoring method and the knowledge of the physics of the failure. High expectations on the availability of the assets stimulate the labor-intensive development of advanced maintenance techniques. The effectiveness of a maintenance technique depends on the practical implementation of made choices, the criticality, and the type of asset (Tiddens et al., 2020).

The first underlying challenge identified by (Tiddens, 2018) is that it is difficult to distinguish between the available approaches. An unclear method results in an ineffective application of available tools and an insufficient understanding of the weaknesses and uncertainties of the model. The second identified challenge is often a mismatch between the ambition level and the required data (Tiddens, 2018). Logically, the expectations created by the demonstrated possibilities with the technology are too high for the limited data that can be monitored of acceptable quality. A third challenge is the difficulty of showing the added value of a maintenance approach (Tiddens, 2018). This may be due, for example, to the residual uncertainty in the model that outweighs the high development costs.

Essential skills

Fundamental essentials to implement data-driven maintenance are already categorized by the maturity models in technological, IT, and organizational factors (Mainnovation, 2018; Van de Kerkhof, 2020). The implementation study of (Van Eijk, 2023) also identified three pillars for a successful implementation of data-driven maintenance: human factors and organization factors, operation technology implementation, and information technology implementation. All essentials describe the same goals summarized in technological, IT, and organizational implementation.

Proactive maintenance transformation framework

The result of designing a practical framework for SMEs by (Van Eijk, 2023) resulted in The Proactive Maintenance Transformation Framework as presented in Figure 12. The framework results from an extensive literature review and is adjusted by reviewing the effectiveness of performing a case study. The framework distinguishes itself by parallelly combining technical, IT, and organizational domains (Van Eijk, 2023).

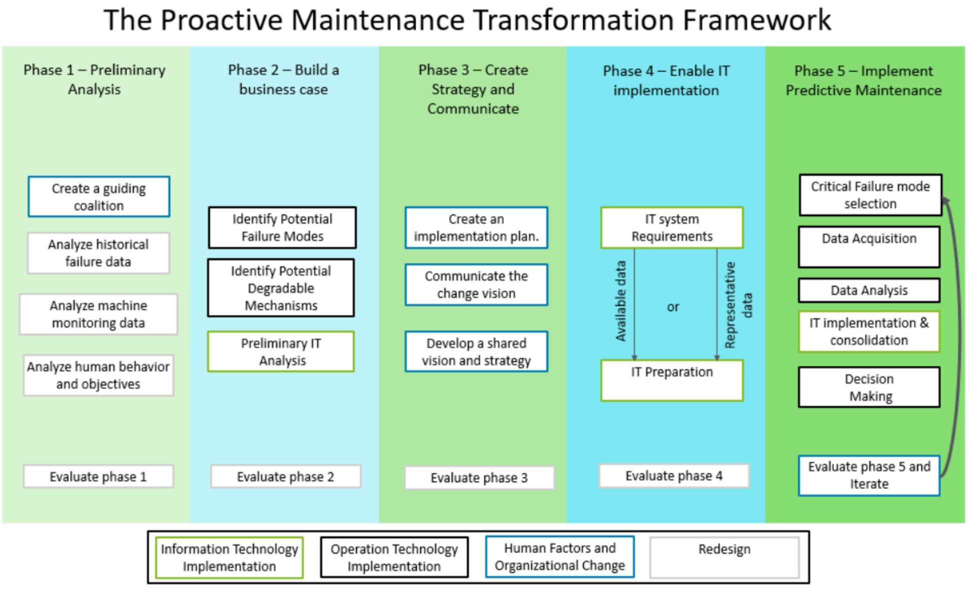


Figure 12, The Proactive Maintenance Transformation Framework for SMEs (Van Eijk, 2023)

Special attention has been given to implementing human factors and organizational change into different steps in the implementation process. Human factors, change management, and stakeholder input are crucial for implementing predictive maintenance (Van Eijk, 2023). Evaluation of all phases must prevent continuing a demonstration disregarding poor results (Van Eijk, 2023). A major underlying message is that barriers are often against change instead of against the implementation of data-driven maintenance (Van Eijk, 2023).

Business case

Data-driven maintenance requires up-front investments in sensors, the IoT infrastructure, and the development of analyses. Adding sensors and monitoring software increases the initial cost of the machine, making the machine less attractive for the customer price-wise. Additionally, part of the OEM's profit comes from selling spare parts, service contracts, and asset checks, so developing smart sensors needs a different approach (Sensorfy B.V., 2023). Some studies show the success of data-driven maintenance. For example, an industrial average from independent surveys by the United States Department of Energy shows a 10 times return on investment, a 30% reduction in maintenance costs, a 75% elimination of breakdowns, a 45% reduction in downtime, and a 25% increase in production (Sullivan et al., 2010). The benefits of data-driven maintenance are reduced maintenance costs, reduced capital expenditure, improved safety/reduced risk, reduced operational costs, increased equipment effectiveness, fewer spare parts, lower warranty costs, and reduced energy costs (Sullivan et al., 2010). Costs and revenues are very context-dependent and, therefore, difficult to quantify.

Privacy, security, ownership

Machine data is gathered, sent to the OEM via an IoT network, and stored in a database. Organizations are concerned about the security of all IoT parts and the privacy of their production data (Sensorfy B.V., 2023). Data sharing between the asset user and OEM can be defined in a legal agreement, end-user agreement, or terms and conditions between the OEM and the client. Stork IMM can use asset data to improve the product, monitor performance, or troubleshoot. In our case, there is mutual consent on data sharing for data-driven maintenance, so the machine data is obtained with permission from the client. As seen in the last decade, improving data sharing conditions and setting data governance requirements are necessary to continue the rapid data transformation (Data Governance Act, 2022). As said, a European regulation 2022/868 has been created, facing the challenge of numerous new appliances. In this project's scope, there is mutual consent on data sharing.

It is important to address the vulnerability associated with trade secrecy. A data-driven maintenance provider can easily derive practical knowledge of the client’s activities from data flows (Druetta, 2018). From the service provider's standpoint, it can be argued that a deeper understanding of the client’s processes can improve the service value (Druetta, 2018). However, it can also be understood from a client's point of view that processing data in the wrong hands can jeopardize their ‘secret ingredients’ in production processes.

3. Research Methodology

The objective is to assess the current implementation process of data-driven maintenance in relation to existing frameworks and academic tools. This chapter investigates action research as a potential research methodology and designs the project methodology.

3.1. Action research

Action research can be suitable for gaining different insights than conventional techniques. To conclude this, knowledge is first gathered about action research.

What is action research?

Development and implementation operations can run smoothly in companies, although implementation projects often do not succeed. Research on the failure rates of change initiatives in organizations shows high numbers, up to 93% (Decker et al., 2012). A trend is signaled toward more integrative research in operation management and suggested selecting different research methodologies (Coughlan & Coghlan, 2002).

For operation managers and academia, action research is a research method that focuses on action points and the execution of a project while at the same time building scientific knowledge (Coughlan & Coghlan, 2002). By the agile structure of action research, replanning and adaption from initial thoughts during the project improves the result. Summarizing a series of publications, (Coughlan & Coghlan, 2002) identified the main characteristics of action research:

- Research *in* action, rather than research *about* action
- Participative
- Concurrent with action
- A sequence of events and an approach to problem-solving

In action research, the researcher participates in a local environment to solve a local problem (Leedy & Ormrod, 2015). Conclusions from action research can be globalized and related to other research. Solving a problem and contributing to science are the two objectives of action research (Gummesson, 2000). The agile nature of action research allows rapid reflections and adjustments for the next cycle. Creating actions in iterative cycles is optimal for creating value in dynamic engineering companies (Humbeck et al., 2020).

By its nature, descriptive research is popular. (Coughlan & Coghlan, 2002) compare descriptive research as positivist science with action research. Positivist science methods generate universal knowledge and validate their findings on logic, measurements, and prediction consistency compared to experimental action research (Coughlan & Coghlan, 2002). While conducting the study in action, the researcher must often take a helicopter view and reflect on how the action fits into the research and hypothesis.

Action research requirements

Every research methodology has certain conditions to be carried out. This section discusses some of the conditions mentioned in reflections from other studies. Firstly, action research requires change. The dynamics of an organization during the project is the knowledge that illustrates the necessity for change and the desired outcome (Coughlan & Coghlan, 2002). In other words, difficulties and obstacles are found best if the organization is willing to change.

The project must be suitable for the iterative cycles of action research. In this project, executing different failure modes can count as the iterative cycles. The project participants need to understand the methodology and the proactive mindset because of the differing research structure.

Action research for this project

Action research is interesting because it can reveal different insights that are not commonly found in traditional research methods. This approach adds a new perspective to the literature on implementing data-driven maintenance in SMEs. Action research is appropriate if the research aims to describe a series of events over time in a group or organization, with particular attention to how and why the action improves a process or aspects in a system (Coughlan & Coghlan, 2002). So, action research is an appropriate method to investigate the implementation process of data-driven maintenance.

The research questions of this study aim to structurally implement data-driven maintenance and discover hurdles for Stork IMM through implementation. The best way to find out is to experience an implementation process. Compared to design research, action research has the additional advantage of its cyclical characteristic. By applying data-driven maintenance we can learn from implementing not just once, but adjust things over cycles and learn again whether it helps.

We can implement a use case in each research cycle to fulfill the iterative cycles. The project is an implementation project that focuses on changing/adding processes in the company. The project is conducted in real-time and a sustainable infrastructure is being sought. All criteria that make action research a proper fit as a research methodology.

However, there are also disadvantages of action research. Action research is known as a time-consuming process. Implementing processes can take much time and depend on multiple stakeholders. The time allocation is managed by executing three use cases and defining them so that this is feasible within the available time. The time pressure is also known among the stakeholders. Informing stakeholders of this research method helps in the success of action research (Coughlan & Coghlan, 2002). A conservative attitude of the company can also cause resistance to change. In this case, it is important to identify why this resistance to change occurs, this may be a generic problem for SMEs.

3.2. Research design

In conclusion, action research is a promising method for this research. We can further design the specific research steps for the research questions formulated in the introduction.

Framing the issue

The issue that needs to be developed in this project falls within the maintenance framework of (Tiddens, 2018) from Figure 9. The difference with the loop presented by PrimaVera (Ton et al., 2020) is the missing link with the asset management plan. This step has not yet been defined because Stork IMM does not maintain the assets or offer a sales model in which predictive maintenance is included. The loop is completed by recommending a maintenance action to the customer.

Before we can start implementing data-driven maintenance, preparatory steps must first be taken. The Proactive Maintenance Transformation Framework from (Van Eijk, 2023) is a recently developed framework to help SMEs implement maintenance techniques based on extensive literature review. The preparatory steps of this framework do fit this project and provides an interesting opportunity to test these preparatory steps. After the first four phases of the framework, phase 5 is ideal for the action research cycles. Three use cases can be conducted to test the framework, ensuring adherence to the action research method. The framework has not yet been thoroughly tested and the real-time implementation in SMEs still shows a lack of examples. The project activities to fulfill the objective of this study are shown in Figure 13.

Figure 13, Structure of the executive part of the research

Action research cycles

The action research cycles defined by (Coughlan & Coghlan, 2002) and assigned for the use cases in this project are shown in Figure 14. The assigned monitoring action for all implementation steps in each cycle reflects the progress of the process. This monitoring should evaluate and improve the effectiveness and functionality of each step for the next cycle.

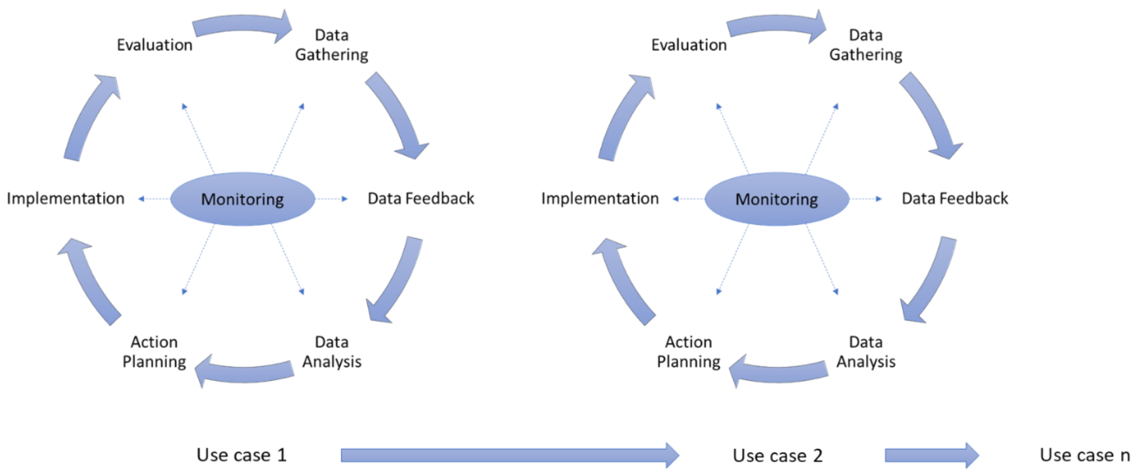


Figure 14, action research cycles for each use case, created based on (Coughlan & Coghlan, 2002)

All cycles comprise six steps: gathering, feedback, analyzing, planning, implementing, and evaluating the data. The meta-step monitoring focuses on the academic contribution (Coughlan & Coghlan, 2002).

Data gathering is the starting step once the use case has been determined. The data-gathering step corresponds to phase B of the presented framework in Figure 9. Before the data can be collected, the use case must be thoroughly understood to monitor the correct data. Since the approach is degradation-based, a deep understanding of the failure mechanics is required, as concluded in the literature review. Next to the hard data from sensors, soft data is collected by the researcher participating in the company during the implementation process (Coughlan & Coghlan, 2002). The softness of data in the implementation processes is due to the researcher's interpretation.

Data feedback reflects on the usability of the data for the analysis. Data science projects are prone to incompleteness or suffering from quality (Sajid et al., 2021). Data analysis is the following step, where the data is used to answer the underlying question and goal of the analysis. It is important to judge the effectiveness of the data collaboratively to include the organization's knowledge and ensure the proper action steps in the next step (Coughlan & Coghlan, 2002).

Action planning is the step where the actions are planned and changes are initiated. Actions to implement the technology are cooperatively determined from input from the previous steps. Types of questions to ask during this phase relate to what needs to be changed, where it needs to be changed, whose support is required, and what resistance is expected (Coughlan & Coghlan, 2002). In this project, we are implementing data-driven maintenance. (Van Eijk, 2023) concluded that there are three main factors in implementing data-driven maintenance: IT, OT, and organizational factors. These categories can structure the implementation actions.

After the implementation step, the result is evaluated. The outcomes of the planned actions are reviewed and the subsequent action cycle must profit from the experience of the previous cycle (Coughlan & Coghlan, 2002). The meta-step monitoring is applied to all steps to ensure the method is followed correctly and to monitor the effectiveness of all steps. Ideally, everyone participating in the project monitors how the steps are conducted and which assumptions are made (Coughlan & Coghlan, 2002). Reflecting on the events aims to conduct the research more effectively. Reflection contributes to an academic outcome that holds universal learning value.

Participants

The project is a student's graduation project in an academic setting. Compared to other participants, the student (the participating researcher) has a substantial fraction of the contribution to the project. The participating roles are described in the next section.

Student – The student is the project leader and main contributor. Apart from the specifically defined tasks of the other participants, the student is responsible for all other activities.

Software engineer – Although the student mainly conducts data science tasks, the participating software engineer implements the monitoring software and other applications in the current machine software. This participant's experience with the sensors is also valuable in effectively monitoring the right data. The role of the software engineer is crucial for the execution, design, and implementation of the required software in the machine.

Mechanical supervisor – The mechanical supervisor advises, checks, and helps the student in the correct mechanical interpretation and execution.

Operational supervisor – The operational supervisor monitors the implementation improvements, guides the company's interests, and helps the student relate academic conclusions to the company goals.

Academic supervisor – The academic supervisor advises on the research progressions and helps relate findings to other research areas. The experience of the academic supervisor also helps to focus on the suitable activities towards the interesting conclusions compared to the current state of the literature.

3.3. Research quality

The previous paragraph discussed how data analysis steps are incorporated into the action research cycle. In this paragraph, the essence of the analysis and areas in which conclusions are sought are described to ensure the quality of the research.

Data validity

The research has two different outputs. In addition to the technical result, we are also interested in the implementation process. Because a data-driven maintenance cycle is conducted in each cycle, the ability of the data to answer the technical goal is evaluated in every cycle. It is important to conduct the monitoring meta-step to gather sufficient information on the implementation questions with this goal in mind.

To maintain validity, action researchers must consciously and purposefully carry out the action research cycles, test their assumptions, and subject them to public testing (Coughlan & Coghlan, 2002). The main threat to the validity of action research is the lack of impartiality on the part of the researcher (Coughlan & Coghlan, 2002). This threat and excessive company influence in steering the project can jeopardize the research's neutrality.

Knowing that validity is especially in danger in experimental design research projects (Leedy & Ormrod, 2015), measures can be drawn beforehand. The technical validity is discussed with multiple in-company experienced personnel to optimize the combination of theoretical correctness and experience in the failure modes. The implementation process is frequently evaluated with internal and external supervisors.

The validity of the analysis can be enhanced by common strategies such as testing a real-life setting, providing other samples, and replication in another context (Leedy & Ormrod, 2015). In this research, solutions can be tested with data sets of different machines of different sizes performing other cycles. Qualitative conclusions can be evaluated by analyzing outliers or contractionary results, providing a detailed description, acknowledging personal biases, or feedback from others (Leedy & Ormrod, 2015).

Execution of the research method

Activities and conditions of action research were discussed earlier in the chapter. In this research it is important to adhere to the structure of action research in order to arrive at conclusions with a methodology that is trackable. In addition to adhering to the research steps, it is important to complete all steps in an acceptable time frame to be able to conduct three research cycles.

The steps of action research defined by (Coughlan & Coghlan, 2002) are designed in such a way that during the steps you focus on the implementation, evaluate the content, and monitor the process. In this study, the quality of the research is assured by diligently following the action research steps so that these properties are guaranteed.

To learn and test different lessons it is important to perform three cycles. The learning curve effect is evident in iterative tasks, where actions become progressively easier with repetition. The learning curve effect means that as you repeat tasks, not only do you figure out what works best, but also successful actions become easier each time you do them again.

Limitations

Due to the flexibility of action research in the implementation process, the generalizability is limited and the research is hard to replicate (Cohen et al., 2017). Generic knowledge can be created by comparing research findings with the frameworks from the literature review.

Another risk for action research is research biases of the research group, such as selection bias, social desirability bias, or other cognitive biases (George, 2023). It is the researcher's responsibility not to let biases limit the research. Taking sufficient feedback from specialists in the research provides the most valuable and innovative results.

Wrapping up

Action research fits the objective of our research questions. Action research can lead to different conclusions about implementing data-driven maintenance compared to conventional research methods. The research method also fits the purpose of this project well. The disadvantages and limitations of action research have been investigated, and various measures should be taken to prevent the research from becoming invalid. The most important measures are sufficient feedback and input from varying experts and stakeholders. The action research cycles presented in Figure 14 ensure robust research and are followed in chapter 5.

4. Preliminary steps for data-driven maintenance

Preliminary steps for implementing data-driven maintenance in Stork IMM are presented in this chapter. The injection molding machine is introduced and a preliminary IT analysis is conducted. These steps correspond to phases 1, 2, and 4 of the Proactive Maintenance Transformation Framework (Van Eijk, 2023).

4.1. The injection molding machine

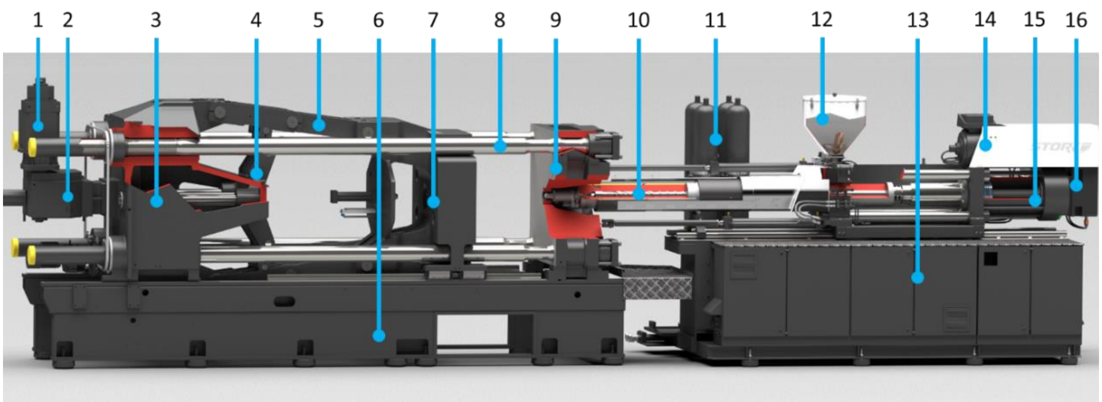
The injection molding machine is super exciting due to its many dynamic aspects and decades of development by many engineers. Many components on the machine perform at their maximum capacity and many technologically impressive solutions are incorporated into these components. This paragraph introduces the injection molding machine for data-driven maintenance.

General working principle

The injection molding machine makes plastic products. A mold is placed in the machine and often a robot is used to eject the products. The machine receives granules in its hopper and melts the granules through friction and heat in the screw, which is driven by the dosing motor. Once the plastic has melted in the screw, the plastic is injected with a pressure of up to 2000 bar into thin-walled molds. The hydraulic injection cylinder takes its pressure from the hydraulic accumulators and fastly translates the screw to inject the shot. Several components mentioned in this paragraph are indicated in Figure 15. All control components of the machine, like the hydraulic pump, filters, and electrical components, are incorporated into the injection frame.

The mold is opened and closed on the left side of the machine in the picture, the closing unit. The mold is attached between the moveable and fixed clamping platens. The electric drive drives the moveable clamping platen via the transmission system with a variable transmission ratio. The electric drive consists of an electric motor and gearbox that drives the crosshead. During injection, the mold must be held closed with an exceptionally high force to keep the high-pressure plastic within the mold. The clamping force is applied by stretching the tie bars and locking the transmission system.

The entire process takes place in cycle times of less than 3 seconds. The cycle time mainly depends on the cooling time and weight of the product. Processes in the machine generate very high loads on the components in many places.



1. Clamp motor
2. Clamp gearbox
3. Back platen
4. Crosshead
5. Toggle system
6. Closing frame
7. Moveable platen
8. Tie bars
9. Fixed platen
10. Screw and barrel
11. Hydraulic accumulators
12. Hopper
13. Injection frame
14. Dosing motor
15. Injection cylinder
16. Dosing gearbox

Figure 15, The injection molding machine

Maintenance of the injection molding machine

Stork IMM helps customers with their service department to keep machines running. There are standard periodic service orders, which mainly involve changing filters, greases, and oils. More extensive periodic maintenance occurs less often but is necessary if machine functionalities are compromised due to wearing parts. All parts that move relatively to each other wear. The most common wear parts are pivots in the lever system, guides for the movable and rear clamping plates, and other linear guides and bearings.

The machine has many potential parts that can fail. What fails depends mainly on usage, the accuracy of the control, and the machining tolerances. In addition to standard wear parts, failures arise due to overloads, incorrect use, or external effects. An example of this is sudden power outages in some countries where Stork IMM machines are running. Customers in these areas regularly experience sudden power outages, leaving the machine uncontrolled at high speeds.

In addition to expected wear parts, there are also less common maintenance parts. These parts do not usually fail. If they do fail, there is often something in the system that causes the failure. External factors include control errors in the software, deviations in the construction, or excessive dimensional errors. Failure of less common parts can cause much downtime due to the longer lead times. Customers with many machines or customers in remote locations often have spare parts to cover such defects. This spare parts set contains tie bars, hydraulic valves, or electrical control cards. Not all parts can be kept in stock in this way. Many parts can cost tens of thousands of euros, making spare parts sets too expensive.

Suitable components for predictive maintenance

Components that are easy to replace and relatively cheap are the most suitable for reactive maintenance, while critical parts in complex systems are more suitable for predictive maintenance (Tinga, 2010). The boundary between the two maintenance strategies is a bit vague. It mainly depends on the ability to monitor the failure mode and a cost-benefit analysis of implementing a data-based technology. We must prevent all failures that lead to more extended machine downtime. Therefore, it is essential to ensure that costly parts with long lead times do not fail. These parts are most suitable for predictive maintenance because the profits are the greatest.

There are currently many sensors on the machine to control the machine. The machine's controls contain functionality to read these sensors every millisecond. Whether we can use these existing sensors to measure the conditions and loads on parts remains to be seen.

In this research, the use cases are provided by active failure cases. These quality-related development projects in Stork IMM are very current, which helps the implementation process.

4.2. IT system

The IT system must provide the facilities to enable data traffic between the sensors and the analysis software to perform the intended data-driven maintenance functions. The IT system is crucial for data-driven maintenance. This paragraph discusses the necessary IT functionalities.

IoT platform

Data-driven maintenance is not new and is a hot topic in many industries, which is why many companies respond to practical demands. Microsoft Azure, Amazon AWS, and Google Cloud Monitoring provide IT infrastructures. Such large companies offer these functionalities in combination with their cloud-based data storage.

There are also open-source IoT platforms. Open-source IoT platforms have the advantage of having a lower entry-level by not paying immediately. Such platforms only charge subscription fees for certain data traffic or storage, making it a good opportunity to break ground on the project and demonstrate its profitability. We performed a test with open-source IoT platform Thingsboard. However, the platform turned out to be more complex than previously estimated. This was mainly due to the specific structure of the communication in the platform and, therefore, tricky problem-solving. The fixed template of the platform also limited the freedom of manipulations and visualizations.

In consultation with the software department of Stork IMM, we have set up our own IoT infrastructure. This infrastructure sends the loggings from the machine to the database via an internet connection. The database runs on a Stork IMM server itself. The steps between the sensor and the final form are shown in Figure 16. The steps are sorted into steps that take place on the machine, steps that take place on and between databases, and steps for the analysis.

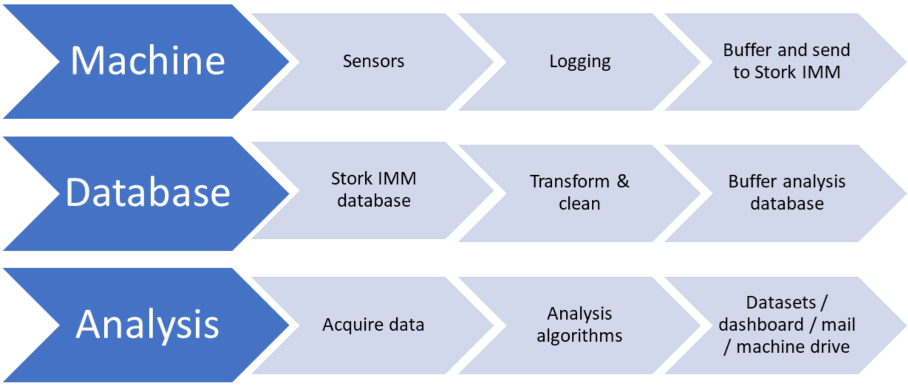


Figure 16, IT infrastructure from sensor to interface

Cloud-based database

An IoT infrastructure has been set up to demonstrate how a cloud-based database works with an external service provider, as shown in Figure 17. It is only a demonstration because the University of Twente facilitates the database storage. The SQL database is used with PostgreSQL database management software. Database administrator phpPgAdmin is also used to store and display the data in a user-friendly manner.

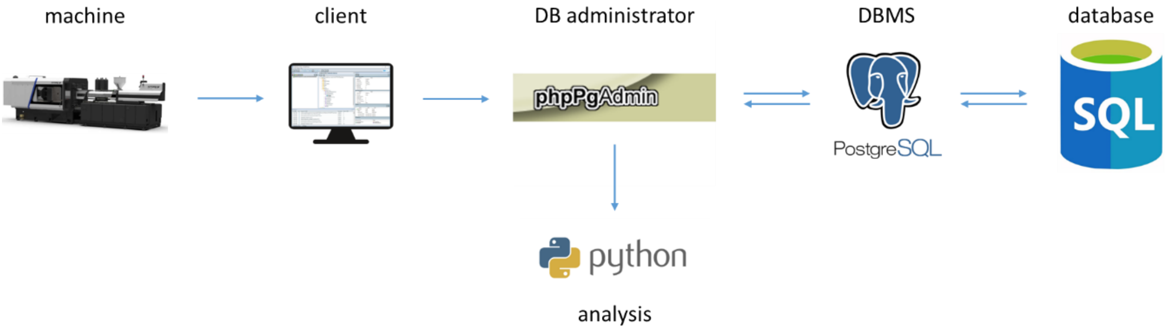


Figure 17, IT infrastructure for cloud database

Visualization

Various systems have been considered to visualize the analyzed data. The desire is to use a visualization program where the underlying code is not immediately visible. Real-time dashboards and business intelligence (BI) tools have been considered. With various tests of real-time dashboards, the limitation in editing freedom quickly arose. The input data must be in a particular format, with little freedom in visualizations and dashboard layout.

Because the editing requirements for visualizing the use cases are high, the decision was to employ a business intelligence tool, such as Power BI or Tableau, to visualize the analyses. All the visualized data is preprocessed using Python to ensure independence from the visualization tool. The business intelligence tool is less real-time but can be updated automatically with the most recent data. The visualization can then be used in, for example, a dashboard, a report, or an email.

5. Case results

This chapter implements three use cases in the company and machines. For the implementation process, we follow the action research activities as described by (Coughlan & Coghlan, 2002) and shown in Figure 14.

5.1. Cycle 1: Tie-bars

Recall the action research steps in Figure 18. After the introduction of this use case, every action research step is presented as a sub-section in this paragraph. The meta-monitoring observations from all steps are combined and shown at the tail of this paragraph.

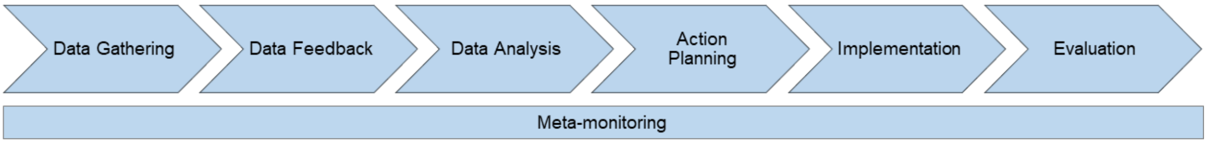


Figure 18, Action research cycle steps

The tie bars of an injection molding machine are crucial in building up a sufficient force to keep the mold closed during injection. The closing force is built up by pressing the mold halves against each other and stretching the tie bars. The four solid steel bars with a diameter of up to 200 millimeters stretch a few millimeters. These bars are stretched each cycle, making the tie bars known for fatigue failure. Tie bar breakage is a known weakness of injection molding machines and can interrupt production if a spare tie bar is not in stock. This scenario is not unthinkable because the price of a single bar can run into tens of thousands of euros. Breaking the tie bars can cause a loss of profits due to unplanned extended downtime. Stork IMM cannot yet predict the breaking of tie bars.

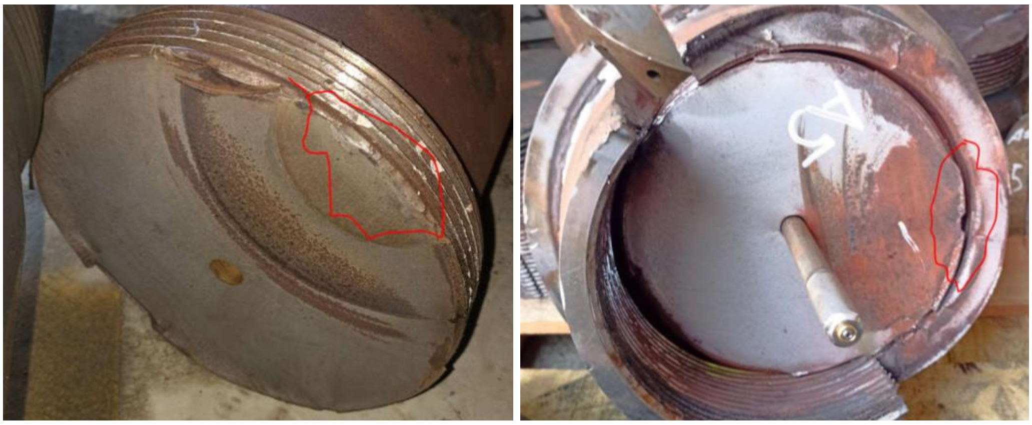


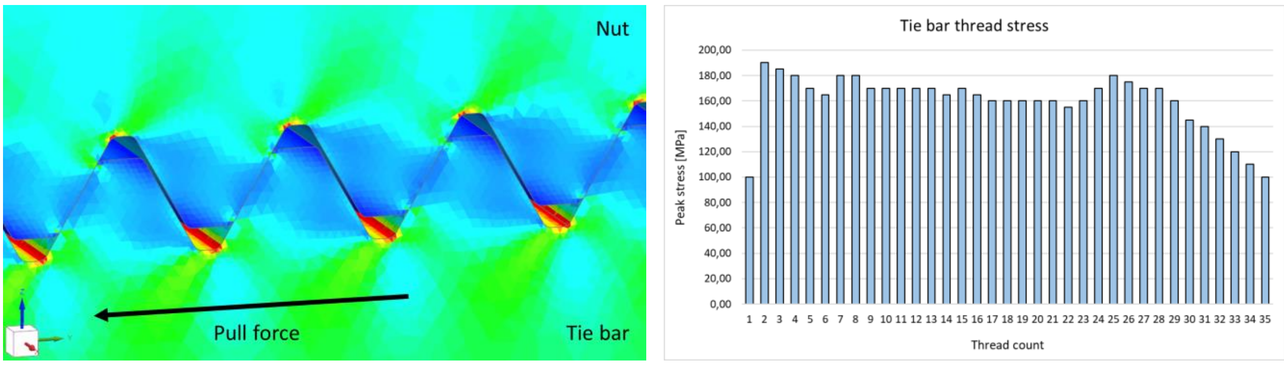
Figure 19, tie bar failures

Step 1: Data Gathering

There are several tricks in designing the tie bars to distribute the force per pitch as best as possible. This significantly reduces the risk of breakage on a single pitch. Perfectly distributing the tension over each pitch is practically impossible. The very close machining tolerances and effects of machining methods cause a deviation between the theoretical design and the practical impact. In the company's experience, there is no way to pinpoint which variables can better predict tie bar breakage. Tie bars sometimes last for decades in critical situations, while other tie rods collapse after four years.

Figure 20, load case of the tie bars

This report examines tie bars with a closing force of 11.000 kN. Four bars collectively generate a tensile force of 11.000 kN. Figure 21 (a) shows the result of a finite element analysis (FEA). It can be seen that a peak stress appears where the tensile force flows into the thread of the tie bar to transfer the force to the nut. Figure 21 (b) shows the peak stress per pitch.



(a) (b)

Figure 21, Tie bar peak stresses

The tie bars contain strain sensors that can measure the tensile force by the strain in the bar. In many machines, all four bars have a strain sensor. The force is calculated using the diameter of the bar in the measurement cross-section.

There are two interesting moments in a cycle to measure the force. The first moment is after locking. This is the built-up closing force for the injection process. The second is during the injection process. The high injection pressure causes the mold to breathe and the closing force to gain. So, four data points are stored twice per cycle.

Step 2: Data Feedback

At Stork IMM, it is known that a short cycle time with a high load accelerates fatigue failure. This makes sense because the amplitude and frequency of the load are increased, both of which are detrimental to fatigue failure. The available sensor can measure both this frequency and load. However, it is known from experience that the fatigue duration can vary greatly, even for comparable situations. This difference is due to dimensional deviations at the nano level and material discontinuities, neither of which are acutely measurable. Due to these deviations, it is not possible to create a generic S-N curve (a curve to set out the stress to the number of cycles).

The tricks used to distribute the stress theoretically are challenging to realize. As a result, the practical effectiveness of these measures varies. This inconsistency leads to a greater standard deviation in the statistical probability distribution. With these expensive components, such a large standard deviation is unacceptable. It can only be used as a guideline for advice on the quantity of spare parts. Additional information about the loads or conditions must be incorporated for a more accurate forecast.

An alarm is set if one of the four rods exceeds a specific deviation. However, this deviation is set relatively large to prevent false positives. For example, notifications can be given if the force in one of the rods is higher due to a non-flat mold. If the tie bar exceeds the set deviation, the tie bar is often already broken.

Step 3: Data Analysis

With current knowledge of the failure, there are two possible degradation models. The first is a fatigue model with the influence of variables. The second is pattern recognition of a reduction in closing force in a tie bar that is starting to fatigue.

Variables influencing the faster degradation of the tie bar include straightness of the load, rotation of the tie rod during life, pitch accuracy, material continuity, and overloads. Most of these are very difficult to measure using sensors. The influence on the standard deviation is also unknown. If the probability distribution has to be designed, it is essential to register the measurable variables properly. However, a reliable probability distribution requires much data on breaking tie bars per machine size. There are not enough failures within construction improvements to establish a reliable probability distribution.

The second concept for a degradation model assumes that the tie bar that breaks carries less force than the other bars. The four tie bars stretch parallel to each other. The total elongation of the tie bars is equal due to the construction. If one of the tie bars starts to crack, it becomes less stiff than the other bars. Because this tie bar strains more, the other three tie bars carry a higher share of the total force. This connection was not digitally observed because only an upper and lower limit of the total force per rod was considered. This type of degradation model does not prevent the breakage, but it can indicate at an early stage that a spare tie bar must be ordered to prevent downtime.

While implementing a data science monitoring project, it is essential to consider what message the data should convey. In this case, the monitored data should answer the following questions:

- How many overloads have there been?
- How are the tie bars loaded?
- Is there a degradation pattern visible?

The analysis algorithms in Python must extract information from the monitored sensors to answer these questions. For each question, consideration has been given to how it can be analyzed, which analysis parameters should be created for this, and whether additional action is required, such as registering overloads so that they can be easily found. All the code algorithms for data manipulations are shown in Appendix D.

Step 4: Action Planning

The necessary actions to operationalize this use case involve reducing maturity gaps in specific areas. The action points required can be classified into three main categories: organizational changes, operational technology, and information technology. These three categories emerged in the literature review as the primary implementation categories for data-driven maintenance. The actions for implementing the tie bar use case are:

- Operational technology
  - Analyzing load and failure mode
  - Tracing and analyzing historical data
  - Design monitoring method
  - Develop prognostics
- Information technology
  - Develop monitoring software
  - Develop data infrastructure from machine to database to analysis algorithms
  - Automating monitoring and prognostics
- Organizational
  - Involve and information stream customer
  - substantiation of required software workforce
  - Communicate operation and use case possibilities

The loads and failures mode must be well understood in operation technology actions to monitor, visualize, and predict correctly. Since 2020, some new machines have included machine logging some parameters. Hopefully, historical usage data consists of a case with a failure of a tie bar. This data can be analyzed to see in advance whether the failure behavior is reflected in the data, whether the hypothesis is true, and to test pattern recognition.

On the IT side, the right software must be developed for the machines within the target group. Logging of some parameters has occurred since 2020, but this data is not immediately available for analysis. The data needs to be accessible by the analysis algorithms and consideration must be given to how these steps can be carried out automatically on machines of interest. We are also going to test the IoT network by transferring data from the machine to the Stork IMM database and further to the analysis dataset.

At an organizational level, things need to change to make implementing this use case successful. It is important to involve the customer in the development to make the result usable and understandable for the customer. Enough support must also be created at an organizational level for the required workforce in the software department. The result must also be communicated to different stakeholders in the company to demonstrate the possibilities and results of the first case.

Step 5: Implementation

Most of the steps that have been drafted as actions have been implemented. However, there are also some problems discussed in this section. Logging of some parameters has been implemented in new machines since 2020. There is only one machine with this logging software, the correct sensors, and a tie bar fracture. This is the only dataset with historical data of the failure. Several problems occurred during the actions to obtain the data from the machine. Due to an incorrect version of remote assistance on the machine, we were limited in accessing the machine. The machine was located in eastern Poland, too far from the OEM to arrive on site quickly. After several attempts with client employees, a Stork IMM employee went to the client. It was noted that this machine has recently received a new control unit. The old control unit on which the data is stored is missing. This time-consuming process typifies the difficulty of data completeness and consistency. Now the failure behavior remains a hypothesis.

Monitoring the tie bars on existing machines in the field is the most interesting. A project is organized for a customer suffering from breaking tie bars in South Africa. This case is interesting because of the long shipping time between Stork IMM in the Netherlands and the customer in South Africa. The outdated software needs to be modified to start monitoring these machines. All IT action steps from the previous section are carried out. Unfortunately, the software update was not successful on the machine. Various errors and crashes damage the machine. After two full days of repairing, the old software is restored, and the machine is repaired to get it back into production. The event confirms the difficulty of adapting existing ‘old’ machines to implement new data-driven maintenance software, especially remotely.

Political and strategic questions slow down processes. For example, obtaining data on damage cases can give the customer thought that the maintenance costs incurred on relatively new machines are defects for which the OEM is responsible. The customer is also reluctant to share data to ensure that the OEM cannot discover that defects occur due to incorrect usage. However, this problem becomes less severe if both parties can agree on the purpose of using the data.

Step 6: Evaluation

The load that creates the failure mode is reliably monitored and displayed. The actions on the technical operations side have ensured that this case was monitored successfully. The IoT infrastructure shown in Figure 16, has made it possible to use the machine's sensor data for visualization. The data is updated daily. Updating the databases, analyses, and visualizations is automated at the touch of a button. It is easy also to automate the intervention of pressing the button by having it performed daily at a particular time.

Monitoring has been worked out in Power BI, shown in Figure 22. The dashboard can be used by customers and Stork IMM engineers to assess how the machine is running. The amount of overloads is displayed. There are relatively few observations compared to the recorded cycles, so the ranges may need to be redefined.

While developing the use case, there was no historical data of tie bar failure. The hypothesis is that one tie bar takes less force compared to the other three. Tachometers that show a percentage deviation compared to the other three were chosen to visualize this behavior. The use of colors has added information about how to interpret the value. This is necessary to convey to someone with less background information, such as the customer, how the values should be interpreted.

Towards the end of the project, a tie bar whose sensors were being monitored broke. The breaking of the tie bar confirms the hypothesis. Figure 23 shows that 3 months in advance, over one million cycles, the breaking tie bar starts a downward trend in force: a crack appaired and it starts to degradate. The tachometers are also in red and many overloads have been measured. All functions of the dashboard show red flags. The next step is to investigate the differences in the gradient of the degradation trend and what the determining variables of this trend are.

To investigate the failure behavior over time, it is also valuable to show the load on the tie rods over a number of cycles. This relationship can be used to observe how the tie bars are loaded and whether a visual pattern can be seen. This is a good check on the automated calculation because employee's intelligence can be added here. Visualizing the pattern is easy by interactively scrolling the range of number of cycles.

Figure 22, monitoring dashboard tie bars

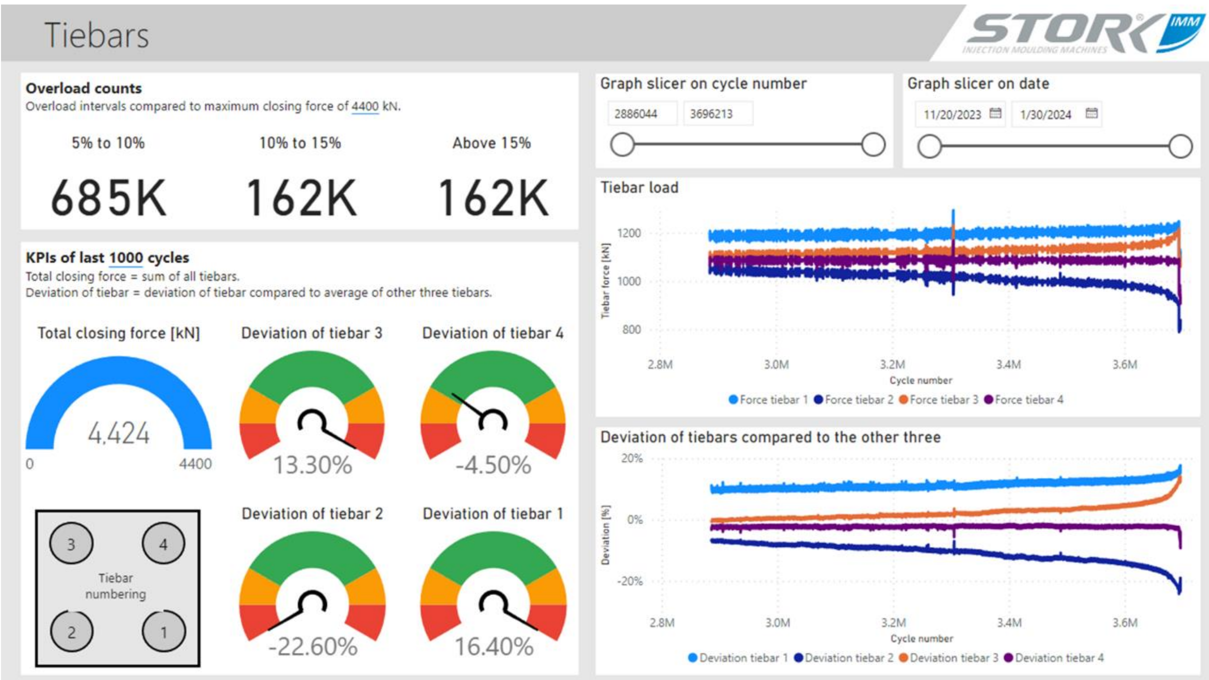


Figure 23, dashboard of a failing tie bar

Meta-monitoring conclusions

The meta-monitoring of each step takes place continuously during the research. The interpretation of observations is discussed in detail in the discussion chapter. This section discusses observations that can be learned from the next action research cycle. After all, the goal of this method is to ensure continuous learning and make implementation more effective each cycle.

After communicating the implementation plan, resulting from the preparatory steps of the Proactive Maintenance Transformation Framework, the first steps of collaboration between the software and mechanical department went more smoothly than expected. The IoT infrastructure was quickly created by combining current solutions and adding a few new functions. This process goes so smoothly that there is some confusion: until now the mechanical department wanted but they found the software inadequate, and the software department did not develop because there is no mechanical demand. The reversal of collaboration hints at the presence of departmental silos, which is evaluated in the discussion under organizational improvements. Good communication about the necessary requirements has quickly resulted in an IoT infrastructure. Communicating the implementation plan with content and objectives provided good direction and indication for both departments, this way of communicating is effective, also for the upcoming use cases.

While working on the project, several tie bars broke in a factory in South Africa. From Stork IMM in the Netherlands, sending spare parts by shipping is time-consuming and expensive via air transport. The situation became so critical that production in South Africa came to a standstill or threatened to come to a standstill on several machines. Predicting tie bar breakage can prevent this situation by responding early by sending spare parts in time or adjusting the production speed to extend the service life as necessary. Due to the situation's seriousness and the project's potential, the implementation process accelerated. Orders arose to supplement missing software and hardware, and time allocation became a priority. So, by carrying out a critical case where monitoring can add value, the implementation process is accelerated due to the high priority of the implementers.

Most of the time in the implementation phase is spent on the IT side. Many issues are recognisable as IT/OT convergence challenges. Creating software, waiting until we can implement it with the right employees, and solving problems took much time. The software department was very busy during the implementation, with understaffing for various reasons, prioritizing other critical work. A key takeaway is to properly plan and communicate the required work of all team members in advance.

Monitoring older machines is the most interesting, these machines have been running for sufficient cycles to be vulnerable to fatigue effects. A remote update of a machine in South Africa unfortunately failed. New software in the old machine caused a frequency controller defect, demonstrating legacy system update issues. Obtaining historical data on a failure mode took much time due to the problems described. The inaccessibility problems teach us not to depend on a single data source. The same can be concluded about the implementation problems of the new software on the old machines. Ultimately, the implementation was successful, but it is beneficial to approach the case to be tested on any machine during the project's duration.

5.2. Cycle 2: Frame load

After the introduction of the use case, all steps of our action research process from Figure 24 are covered in separate sub-sections. The paragraph ends with the meta-monitoring observations of all steps.

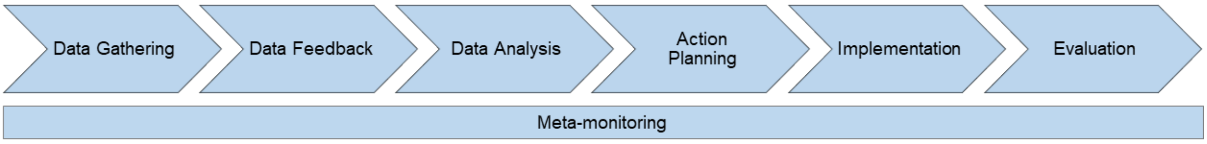


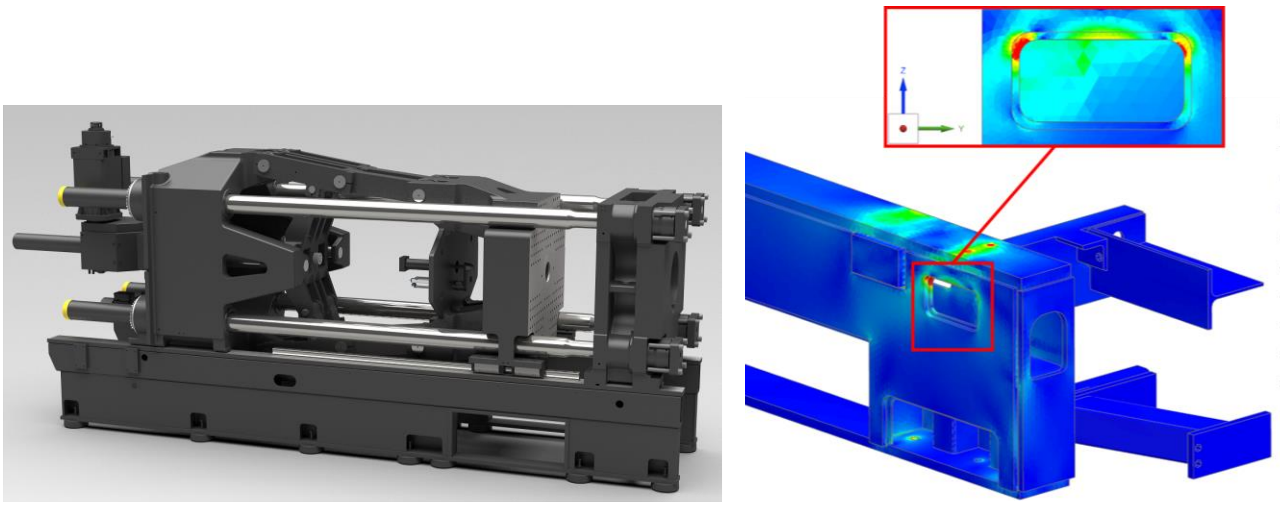
Figure 24, Action research cycle steps

Cracks in the frames of injection molding machines in Australia initiate the following use case, shown in Figure 25. Several cases of cracked frames have arisen in the history of Stork IMM. Therefore, the frame can be a vulnerable part in specific cases. Certain situations cause frames to crack. In these situations the frames are subject to higher loads in certain places. An example is when the frames are glued to the floor. This extra fixture changes the force pattern in the frame.

Figure 25, Cracks in the frames

Step 1: Data Gathering

In this specific use case, holes have been made in the closing foundation at a place where the internal stresses have to pass during an overload. This overload is caused by occasional hard braking for, for example, an emergency stop or incorrect operation. The fixed tension plate, shown in Figure 26a, is the only part to transfer this shear force from the drive to the frame. Figure 26b shows the peak stresses where the cracks occur with a FEA calculation.



(a) (b)

Figure 26, Closing unit frame (a), FEA of load around hole (b)

Some of the machines at risk of this failure are in poor condition and are being replaced. Some frames only have minor cracks or none at all. Monitoring the condition of these last frames is very valuable. Doing this can prevent further cracking and save a lot of maintenance costs and downtime.

The internal stress, which is the load in the material causing the cracks, repeats itself every cycle. However, overload situations, such as emergency stops or incorrect manual operation, can significantly elevate this stress. An acceleration of the LSP causes the load. This LSP is driven by the crosshead via the transmission system. The transmission system has a varying transmission ratio over the entire crosshead path. The drive directly controls the crosshead. The reaction force of the drive is transmitted via the APL and the tie bars to the fixed clamping plate, which ultimately transfers it to the frame. An accurate position sensor on the crosshead and knowledge of the transmission allow us to monitor the acceleration of the drive and the LSP. We can also monitor the force of the drive with the pressure sensors in the hydraulic drive cylinder.

Step 2: Data Feedback

A possible solution for monitoring the condition of the cracks is to place strain sensors around the cracks. This makes it possible to monitor whether the internal tension in the material decreases in proportion to the load on the machine. If the load around a crack decreases, it can be assumed that the crack has ruptured further. The disadvantage of this method is the complex comparison and it only tells something about a specific point around a single crack. The method is also expensive, vulnerable, and not yet proven.

Using existing sensors to determine the forces on the frame is a faster method. These forces also say more about the load on the entire frame than the condition at one specific point. When determining frame load, the LSP's acceleration and the drive cylinder's force are the two best indicators of the force on the frame. The driving force is determined by measuring the pressures on both sides of the piston and multiplying by the area of both sides. The acceleration of the LSP is calculated from the position sensor of the crosshead. The non-constant transmission ratio calculates the position of the LSP. The acceleration can be determined by differentiating twice over time. Because sensor data is not continuous but discrete, noise is unavoidable. The data is usable through filters in each intermediate step. In the experiment shown in Appendix C, an exponential moving average (EMA) filter was used in each intermediate step. With this type of filtering, the data's amplitude is preserved as much as possible.

Both sensors are read every millisecond in the cycle. With a cycle time of, for example, 8 seconds, it is self-evident that storing 8000 data points per sensor is not feasible. The challenge is to collect sufficient information with as little data as possible. Two options have been devised for this problem. The first option stores the minimum and maximum values during opening and closing in predetermined crosshead zones. The second option stores the maximum and minimum values with the corresponding crosshead positions when opening and closing. To determine which method generates the most valuable data, testing revealed that while the first option yields more data, the same conclusions can be drawn from the second option with fewer parameters logged per cycle.

Logging is only done in automatic production mode and during special events. A preliminary analysis concluded that an overload can occur in manual operation if incorrect operation is performed. The machine can be closed and opened manually by a control button. A safety standard stipulates that the machine must stop within a certain distance after releasing the button. Suppose the operator has not adjusted the operating speed and releases the button before the end of the movement. In that case, an emergency stop can be performed when the levers adopt an unfavorable gear ratio between the drive and the clamping plate. This malicious event should be logged to monitor if the frame is being overloaded. For this reason, the logging software must be adjusted to log in manual mode.

Step 3: Data Analysis

An interesting discussion arose between several engineers about the sensors used. The force of the drive cylinder and the acceleration of the LSP are indirect measurements. This does not show the specific force on the frame. The variable gear ratio also makes interpretation of the data more difficult. Recalling that logging aims to map the force on the frame, we have concluded that with these two data sources, all foreseeable possible scenarios are monitored. An emergency stop will be seen as a pressure peak in the drive cylinder. A too-aggressive cycle is reflected in the acceleration of the clamping plate.

However, we now only monitor the highest and lowest accelerations and forces during opening and closing. If an outlier is monitored, we have no information about the cycle other than the measured position of the outlier. It is also unknown whether this is the only high outlier or if there are several high outliers in the cycle. It is also possible that a more harmful outlier is hidden by a measurement error or other recorded value.

Step 4: Action Planning

The action points arise from the experience and lessons of the previous use case combined with the work that still needs to be done to implement this use case. The action points are divided into the 3 categories required for data-driven maintenance:

- Operation technology
  - Load and failure mode analysis
  - Test saving the right data
- Information technology
  - Designing monitoring software
  - Analysis algorithms data
  - Visualization of data
- Organizational
  - Creating substantiating workforce dedication software department
  - Creating implementation plan
  - Communicate working and possibilities

What we want to measure in the closing drive is a complex event. An overload can have different causes and can be measured on different sensors. Other limit values also apply at any time in the cycle. The measured sensor values must be carefully analyzed and we must ask ourselves whether the measured data clearly reflects the harmful load.

The overload only happens for a fraction of a second and we cannot store too much data. The data-gathering method must be carefully considered, developed, and tested. In addition, the measured data must be visualized. This use case requires much work from the software department to create software for machines that monitor and send the data. This software must also be installed on the machines under interest.

Step 5: Implementation

The implementation is most effective with a concrete and relevant project, as experienced in the previous use case. This use case concerns machines in Australia. An action plan has been created within this project with a concrete assignment for the software department. However, understaffing in the software department has caused delays in the implementation process. The software department handles many jobs for the service department, where machines stand still in the field and wait for the required software. These jobs were automatically given priority due to the high pressure from the customer. After the software for this project was created, it could not be implemented immediately due to the risk of required readjustments on the machine and not having access to the staff who can solve this issue.

Furthermore, during the implementation process, it turned out that the interpretation of the data was complicated. Unlike the previous use case, meters with a green, orange, and red color scale could not be used. The scatter plots in Figure 27 show what is happening, but a lot of domain knowledge is required to interpret them correctly. Analyzing overloads becomes easier by specifically displaying overloads with the cycle before and after the overload. The dashboard is more of a tool for specialists. An image with the position to display the sensors has also been added to speed up the interpretation.

Step 6: Evaluation

Despite the complex event, a brainstorming session with engineers from different disciplines led to the development of suitable sensors to monitor the event. Developing software that provides as much information as possible with as few data points per cycle has been achieved by storing the maximum and minimum force and acceleration with the corresponding crosshead position for each back-and-forth movement. This method provides information about the maximum load and acceleration at a certain point with 16 data points per cycle.

In addition to the overload amounts, the visualization shows the distribution of the maximum and minimum forces and accelerations with associated positions. An engineer can assess how the machine is performing. To make it easier to analyze the overloads, cycles before and after the overload cycle are shown. This way, some context is provided to the event, speeding up the analysis process. The dashboard is shown in Figure 27.

Figure 27, Dashboard frame load

Meta-monitoring conclusions

The previous and this use case concern maintenance cases where reliability is increased. However, these cases also depend on the failure of a product to be proven. Proving reliability-related maintenance cases can therefore take more time. It would help if we could quickly demonstrate value to show the profits of the investments. For the following use case, it would be good to look at machine optimization with data such as productivity improvement or energy consumption manager.

In interim discussions, management expressed interest in showing visualizations. Displaying visualizations helps keep management and team members on board and quickly share and expand the potential of the technology.

This use case does not include a forecast. The dashboard purely shows the load on the frame. However, this functionality was very well received. In the past, such damage cases were often attributed to machine malfunctioning. With this monitoring software, the suspicions can be confirmed by extracting much information about the context from the data.

While carrying out this case, the technical content sometimes became quite complex. Unlike the first use case, the load on the frame is not directly measurable in the second use case. In the implementation process, this led to discussions about correctness, the use of sensors, and uncertainty about the technical content. This difficulty is also the reason why the dashboard is hard to interpret. This has been improved by adding images and more context about outliers. After several brainstorming sessions with different departments, several points often came back to two things: Make it easy for the customer to use and use the technology for immediate improvements. This feedback again points to using the technology to optimize the machine.

Despite the planning and specification of the activity as a learning lesson from the first use case, the IT activities were again an inhibiting factor. New required software, understaffing in the software department, other projects with higher priorities, and waiting time for customer actions are a few causes. Ultimately, through various tests, software was developed that provides as much information as possible about the magnificence and location of the highest loads with a low amount of data points per cycle.

5.3. Cycle 3: Machine optimization

In the previous use cases, the value is best seen when something fails. To quickly demonstrate the system's value, there is a desire to conduct a use case that utilizes monitored data to improve machine performance. Recall the action research steps, presented as sub-sectoins in this paragraph:

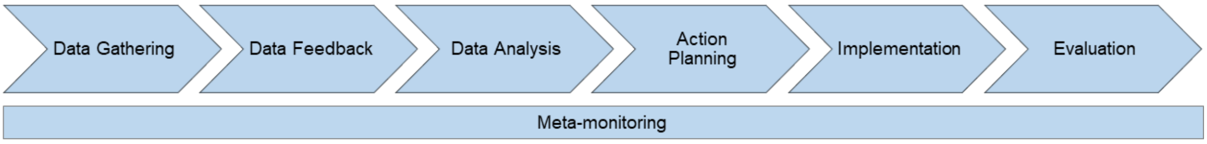


Figure 28, Action research cycle steps

Step 1: Data Gathering: Energy usage

Energy usage is a hot topic. At Stork IMM, customers also want to minimize energy usage. In this way, customers reduce running costs and thus the total life cycle costs of the machine. With high fluctuating energy prices, the injection molding machine's energy use is an important KPI.

A good application would be to help the customer estimate and reduce energy consumption. If we monitor energy consumption and calculate it theoretically based on the setting parameters, we can compare the calculation with the measured values. In this comparison, we can even apply AI to identify the deviations of the calculations on different data sets and thus make the calculation very accurate. The result of this use case is that a theoretical simulator has been created with which we can accurately predict energy consumption. We can use this simulator to achieve the lowest possible energy consumption within the customer's limits, such as a specific time for production. It also has great value for Stork IMM. If we know the energy consumption per drive, we can also see where we can improve energy consumption.

Collecting data with the current monitoring solution is not efficient. We currently collect the selected parameters every cycle. Significant amounts of data is created if we monitor the setting parameters and energy consumption every cycle. This is also unnecessary because the machine produces virtually identical data cycle after cycle. A suitable solution is daily logging, which logs data from a single or average cycle. Unfortunately, we have estimated that the amount of labor for this new software functionality is not feasible within the scope of this project.

Step 1: Data Gathering: Machine productivity

Feedback has repeatedly emerged that customers may have a greater need to prevent downtime than complicated software solutions such as these. Hence, the idea arose to monitor productivity and downtime to monitor the machine's productivity.

There is much potential in the data already monitored up to this case. All kinds of information can be extracted from this data. KPIs such as the number of products, cycles, uptime, downtime, scrap rate, failure causes, and OEE (overall equipment effectiveness) are interesting for displaying productivity. We can use the current loggings for these parameters. Currently, the start time and product counter are logged for each cycle. Alarms are also logged. Combining these two loggings can provide information for the previously mentioned KPIs.

Step 2: Data Feedback

Extracting data from the available data requires several calculations and assumptions through the indirect measurement of KPIs. This accuracy can be greatly improved by implementing the daily software logging functionality as proposed in the previous section. A daily log can contain parameters that draw more accurate conclusions. However, the available data can serve as a demonstration and provide quite a few interesting conclusions.

Step 3: Data Analysis

From the standard logging files, the number of rows can be related to the number of cycles, the number of products can be read, and the cycle time can be iterated by the start time of each cycle. The cycle time can be calculated by calculating the time between cycles. At times, the machine remains stationary for longer periods between cycles, so only cycle times under 40 seconds are included in the average. Otherwise, it is not considered a cycle.

All machine alarms are logged in a separate file. However, automatically analyzing the machine status based on these loggings proved difficult. The inconsistency and incompleteness are both very significant. This is caused by the status not always displayed depending on the type of alarm and by displaying multiple alarms during some standstills. Various samples also have compatibility problems between the cycle and alarm logging files. Due to the inconsistency, incompleteness, and compatibility issues, combining the current alarm and cycle loggings for an automated analysis algorithm is impossible.

The alarm logs can be adjusted to monitor the number of failures per alarm. Alarms that do not cause failure, alarms that say something about continuing production, and alarms that follow the same cycle as a previous alarm have been filtered out.

Step 4: Action Planning

No new logging software is being developed for this use case. However, it is necessary to investigate what information can be reliably extracted from the current data. Because this use case serves as a demonstration, no customer is involved in the implementation and organizational actions are limited. The action points are:

- Operation technology
  - Determine productivity KPIs
  - Convert data to operational context
- Information technology
  - Design analysis algorithms
  - Develop dashboard
- Organizational
  - Communicate results and possibilities

Step 5: Implementation

Apart from the already known limitations of being unable to link the cycle with alarm loggings and the inconsistency and incompleteness problems, developing the case was relatively straightforward. Thanks to prior experience in communicating between Python and PowerBI from previous use cases, as well as experience in integrating separate analyses into the algorithm, incorporating this use case was performed relatively quickly. The information from this use case is well-known in the industry, resulting in a more straightforward interpretation. The data and dashboard also require less manipulation to facilitate interpretation.

Step 6: Evaluation

KPIs have been calculated from the available data about the amount of production numbers and time per day. Information about the number of stops for a specific alarm has also been extracted. A functionality that would be a good addition is linking the number of daily stops and the associated downtime per error, which can be achieved with the daily logging function. The KPIs in percentages are estimated with assumptions, but they can also be improved by implementing time-counting parameters in the daily logs. The productivity dashboard can be seen in Figure 29.

Figure 29, productivity dashboard

It is valuable, for example, to monitor the amount of downtime per cause of downtime. The logging software must be adjusted for these KPIs. Developing a daily logging functionality is the most accurate rather than making the current alarm and cycle logging complete and consistent. Maintaining time counters in the software per alarm or machine company is straightforward. This makes the result more reliable and new interesting conclusions can be drawn that directly helps the customer to reduce downtime.

Meta-monitoring conclusions

While exploring the use case for simulating energy use, brainstorming took place with various departments. Interesting input and feedback emerged during these sessions. This functionality was seen as a step towards a self-adjusting machine that can remove much of the complex understanding of the process from the customer.

Questions about how this solution can be sold to the customer arose. Ultimately, the industry is all about developing solutions that can be sold to generate revenue. Discussions arise as to whether these functionalities are sold as separate systems or applications or whether it is a condition for selling machines in the future. For example, the latter can be seen in the automotive world, where functionalities like adaptive cruise control or lane assist are necessary for cars to compete on the market.

The third use case contains an optimization of the machine with the same IT infrastructure as the previous use cases. Optimizing the machine has faster feedback in value to show the potential of the techniques. The productivity improvement is also very recognizable for users, making it easier to apply and learn from. This case was quickly realized using the current IoT infrastructure and data from already monitored data. The case demonstration quickly found new applications by decision pulls instead of a technology push, such as recognizing machine power outages.

While demonstrating this new functionality, new applications quickly emerged. The same analysis has been used to analyze whether there have been machine power outages.

6. Discussion and Implications

The discussion aims to interpret the study results, relate them to the research question, place them among other studies, and point out limitations (Leedy & Ormrod, 2015). To do this, the findings are interpreted. With the interpretation of the findings, a framework is constructed to help future implementers with the lessons learned in this implementation process. The chapter ends with implications, a reflections on the research method, limitations, and suggestions for further research.

6.1. Interpretation of findings

Instead of a chronological order of events as presented in the experiments part, conclusions in this section are grouped by topic. These topics describe themes that give us insight into the research questions.

Condition monitoring

The literature review focused on condition-based maintenance, industry 4.0, and data science to investigate how data-driven analysis techniques can be used to monitor the condition of components from the injection molding machine. The approach to determining the condition is divided into measured or calculated conditions (Tiddens, 2018). The condition can be measured by directly monitoring the condition or structural health, or a physical model or data analytics can calculate the condition. The models can approximate the condition, but (Tinga, 2010) indicates that measuring the condition, load, or use has incremental steps of residual uncertainty in the analysis. In the first use case, the load could be measured directly. In the second use case, this was iterated from other sensors. The additional calculation model to visualize the load introduces uncertainty due to the indirect measurement. The required calculation models have been demonstrated by (Tinga, 2010) and shown earlier in this report in Figure 6. Measuring the load or condition directly, if feasible, is beneficial for reducing uncertainty and work time.

The study did not perform an FTA or FMECA to identify components in the injection molding machine that could cause critical downtime. For the use cases in this research, well-known quality problems on specific parts were used. This guarantees that the use cases are relevant. Not having to perform an FTA or FMECA to find critical components indicates that the need for data-driven maintenance is indeed there.

Predictive maintenance

Monitoring the condition is a first step, but to improve reliability, this condition must be used to make a forecast. The most ideal outcome of a prognosis is remaining useful life (RUL) or time to failure (TTF). The prediction is based on a statistical failure distribution or a physical degradation model (NEN-EN, 2019). The first use case resulted in monitoring the load on the tie bars. The load that causes the failure mechanism can be measured reliably and accurately. The dashboard has been adapted to the expected degradation pattern and is easily interpretable using color scales. The failure mode of the tie bar breakage is a decrease in strength in one of the four bars. (Tiddens, 2018) indicates the necessity of historical data for making a prognosis earlier in Figure 10, we experienced the same. There is a clear hypothesis about the relegation pattern, but a prediction was not possible because the time frame of the degredation was not known. Ultimately, we succeeded in diagnosing this degradation pattern. Multiple diagnoses are now needed to understand the differences in the gradient of the degradation trend and what the determining variables of this trend are.

The reliability of Stork IMM machines can be improved by using conclusions from data-driven maintenance techniques for maintenance decisions. In this project, we showed that usage, load, and condition monitoring can support maintenance decisions to improve reliability. In the second use case, the result was not a forecast but a load visualization. Despite missing the prognosis step, the result was received as very useful. Visualizing the load avoids suspicions in discussions and helps to understand and control the root cause of the degradation. This indicates that with data-driven maintenance, the result does not always have to include a forecast to be valuable. For a company, it is important to determine the required ambition level of the data-driven maintenance technique. The Maintenance Techniques Framework (Tiddens, 2018) distinguishes decisions between detecting, diagnosing, and prognosing.

Implementation tools

Several data-driven maintenance frameworks were shown in the literature review. The methods generally follow the same steps but differ in specific focus points. The corresponding steps are: Critical failure mode selection, identifying degradable mechanisms, data acquisition, data analysis, evaluation, and decision-making. If one seeks guidance on implementing data-driven maintenance, one promptly encounters what is commonly called maturity models. With these maturity models, maturity can be assessed on key capabilities, divided into technological, IT, and organizational factors. The models are based on the idea that all defined criteria must be around at least a certain maturity to achieve a certain data-driven maintenance level. Examples of maturity models from (Kerkhof, 2020; Mainnovation, 2018) are shown in Appendix B. How it is presented already suggests that a prerequisite for data-driven maintenance is specialized skills across various domains, especially technical, IT, and organizational aspects.

An implementation framework was designed by (Van Eijk, 2023) based on a thorough literature review of many data-driven maintenance frameworks and known challenges. The framework has the following steps: Preliminary analysis, building a business case, creating and communicating strategy, enabling IT implementation, and implementing data-driven maintenance. Instead of directly implementing data-driven maintenance, the framework focuses more on the IT basis, project management, and organizational changes. These basic steps are crucial for SMEs before data-driven maintenance can be implemented.

IT/OT convergence

The IT system must provide the facilities to make sensor loggings available for analysis. Because the storage of the data and its analysis do not take place at the location of the injection molding machine, an IoT infrastructure has been designed. We looked at using commercial solutions for the IoT infrastructure, such as Microsoft Azure or AWS. However, these programs' specific structure and communication led to compatibility problems with the Stork IMM machine software. Other functionalities quickly require add-on packages, and there is also a financial hurdle due to the monthly subscription fee, delaying the project's profitability. Because third-party platforms often use a specific format and are behind a monthly payment barrier, we have created an IoT infrastructure running on Stork IMM servers that is well-compatible with the machine control software.

A trial with an open-source platform demonstrated a lack of freedom in manipulations and visualizations. The open-source test encountered challenges with the more complex-than-expected communication infrastructure, causing time-consuming and frustrating processes due to its initially perceived simplicity. Limited understanding of the communication structure and the communication code being hidden from the user made problem-solving harder.

With both the commercial and open-source variants, it turned out to be very difficult to integrate the company's OT processes with the IT infrastructure of the programs. Many applications and systems have been developed within Stork IMM, each fulfilling a function. Switching to a new system, such as a PDM system, is very intensive with the limited capacity and required functions. The subsequent challenges, such as connectivity and data collection, are only future challenges. The IT/OT convergence is incredibly complex with existing systems.

Another IT/OT convergence issue occurred during an unsuccessful software update in an old machine. Implementing new software functionalities on old hardware caused follow-up issues in this case. When updating legacy systems, problems are created by the technology, architecture, and functionality of the application (Yokogawa, 2020).

Data completeness, consistency, and availability

Sorting out data was often very time-consuming with regularly disappointing conclusions. Problems with completeness, consistency, and availability were common in existing data from the machines, which are typical data science challenges (Sajid et al., 2021).

It also became clear that asking for historical data is unusual. Several challenges arose when trying to get historical data for the use cases. This shows that software on machines at various customer locations was not yet designed to retrieve historical data. For example, data has been deleted, machine software is unsuitable for remote access, and there were problems with connectivity.

It is good to build up data from the machines. The loggings require little storage memory at 350 kB per day, logging 141 columns per cycle. The last use case demonstrated how much additional information value can be extracted from data as long as the data is there. The data we monitor is minimal, exemplified by Formula 1, where one Formula 1 car produces 1,5 terabytes over a race (Shapiro, 2023). Another example arises from a Boeing 787, monitoring 1000 parameters continuously leading to 20 terabytes per engine per flight hour (Badea et al., 2018).

Interdisciplinary collaboration

data-driven maintenance requires modern skills from different areas of expertise. Missing professional employees like specialists and data scientists is a common risk in implementing data-driven maintenance (Mesarosova et al., 2022). The technical content is complex, and the required IT facilities are modern. Implementing data-driven maintenance at Stork IMM requires mechanical, process, and software engineering cooperation. So far, it has not been successful because the mechanical engineers do not find the software goal-achieving and usable. In contrast, the software engineers did not continue the development of the system because there was no expressed mechanical need from the engineers. It's a textbook example of organizational silos, which results in a lack of cooperation that reduces the potential of the project (Yokogawa, 2020). Cross-disciplinary actions are not easy to implement. Defining tasks and waiting for each other in the meantime makes collaboration difficult. The agile approach of this research can be a good example for organizing the project management of these projects.

Creating value

Before a data-driven maintenance use case creates value, it depends on the occurrence of an event and how valuable the output information is to make a decision. The downtime that can be prevented is precious, but the time between the implementation project and the occurrence of the value return can be long. During the tie bar fracture project, the use case became very relevant for a customer in South Africa due to the shortage of spare tie bars. When sending new spare parts, there was discussion about sending some of the new parts by plane to be on the safe side, which is, however, more expensive than sea shipping. If the use case had been further developed and it was possible to monitor these machines, air transport costs could have been avoided. In this situation, the cost savings can be quantified.

An optimization case was carried out for the third use case to achieve faster returns from the application. Directly optimizing the machines is a recognizable cost-saving for producers. It also brings the Stork IMM's optimizing knowledge into the machine, allowing the manufacturer to use this knowledge to get more out of the machine. This is a competitive advantage, like a fuel consumption monitor or lane assist in cars.

Criticality of use case

External factors can determine the criticality of a project. This defines how much priority the work has for the personnel and the progression of implementation. This can make a big difference, especially for SMEs with limited capacity. For example, with the problem of breaking tie bars, the use case became very topical. Implementation gained momentum because development tasks were given priority. So, finding relevant use cases is very important.

Conversely, implementation can also be postponed. During the second use case, the occupancy in the software department was reduced due to standard staff absences in companies due to illness or vacation. Critical projects, such as machines at a standstill, were automatically prioritized.

Other organizational factors

In addition to the conclusions in the previous paragraphs, which can also be placed under organizational factors, there are other organizational and policy aspects to evaluate. For example, it sometimes proved sensitive to involve the customer in obtaining data to discover failures. We did not want to give the impression that we were afraid of a failure. However, maintaining high machine availability is in the OEM and the production company's best interest. Therefore, finding common ground on the data's purpose is a practical solution.

There were also times when we depended on customer actions. For example, we depended on updating or changing settings on the machine. This dependency brought intercontinental remote challenges and complex planning.

The third use case was conducted much faster than the first two because it could use the current IoT infrastructure, analysis algorithms, and other systems. Therefore, The high setup costs only occur initially and when new functionality is needed. In addition, it was good to communicate results and possibilities with leaders from different departments occasionally. Not only to justify spent labor hours and steering the application, but also because it provided new ideas and cases for the project.

Artificial intelligence

Artificial intelligence is used a lot these days due to the demonstration of its application by natural language processing (NLP) applications such as ChatGPT. AI can also show its value in predictive maintenance. An obvious example is that AI can recognize failures based on training datasets of failures and regular cycles. A requirement is that there are sufficient training datasets. So we have to wait for several failures until an AI algorithm can be trained to predict for example tie-bar failures. The word AI also fell in the energy monitoring use case. By comparing the energy use of cycles with the calculation, AI can find patterns in deviations in the calculation. In this way, Stork IMM simulators can be optimized. This is a step towards a self-adjusting machine, like an autopilot for operating the injection molding machine.

6.2. Agile and Modular Implementation Roadmap

We managed to implement data-driven maintenance in the SME Stork IMM. To transfer our most important findings, this section presents a roadmap/framework with the message we would like to give future implementers of the data-driven maintenance technology. The roadmap contains the most representative and knowledge-transferring frameworks according to the implementation process of this research. The roadmap also shows the relation between frameworks and adds action points for improvement that emerged from this research. To not reinvent the wheel and not add another new framework to the list, a selection of the many frameworks from other researchers is used to guide the reader through the current state of the theory. An important conclusion is that the frameworks all have an individual message and focus but pursue the same goal.

One reason this project's implementation of data-driven maintenance was successful is the action research research method. The research method allowed the technology to be implemented in an agile setting. This also made it possible to improve complexity, capability, and automation in subsequent cycles. In literature, agile work is linked to providing flexibility, exploiting technology, and facilitating change-oriented behaviors (Cimini et al., 2024). The roadmap has a cyclical property to modularly increase the skills, abilities, and complexity in the cycles. References to use here are maturity frameworks. These frameworks indicate the skills required at a certain ambition level. This step can be seen as the loop in the simplified view of the roadmap in Figure 30 as *increasing skills and capabilities*. The other steps are explained after the figure.

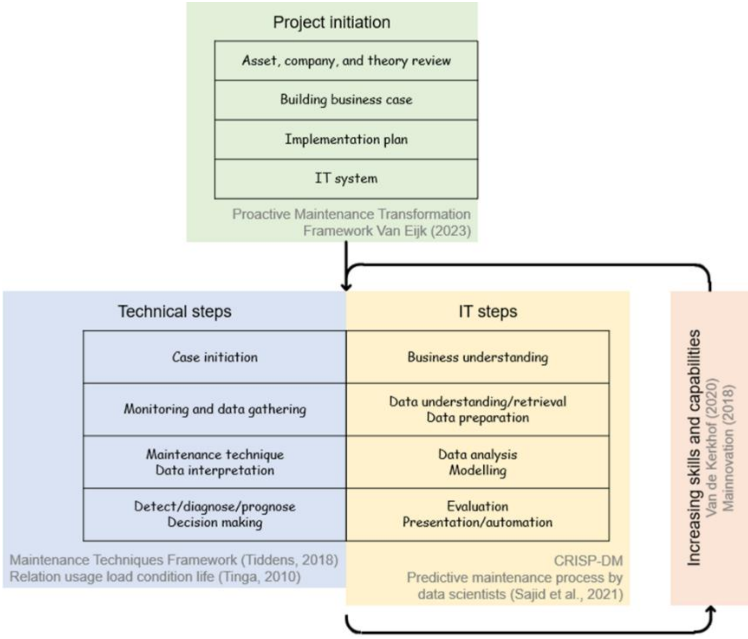


Figure 30, Simplified version of the roadmap

The preparatory steps are mainly used from The Proactive Transformation Framework of (Van Eijk, 2023). This framework has been compiled based on extensive literature research and the needs of SMEs. In this study, the preparatory steps of this framework were conducted and served as a good basis. Doing a proper literature review of the techniques for implementing data-driven maintenance is essential. The preparatory steps should also facilitate an IT infrastructure that suits the company and enables data monitoring and analysis for the first application cycle. The framework is partly focused on the organizational side of changes. From the start, this research did not aim to change business processes directly but to develop a tool that is increasingly used to fulfill a business need. The improvements to these preparatory steps have been implemented by adding steps that look at the business needs that the tool highlights and determine the final form.

In applying cases, there is a clear relationship between IT and technical steps. Frameworks on both aspects go through the same process in similar steps. This roadmap lays out various frameworks parallel to see the expected steps in the IT and technical fields. Placing them side by side shows the relationship between the frameworks and facilitates ease of use by providing more context in fewer frameworks. The general steps can be seen in Figure 30, and the specific steps of selected frameworks can be seen in Figure 31. The frameworks are reused as a guide through the existing theory.

The Maintenance Techniques Framework (Tiddens, 2018) is used for the technical steps. This framework shows techniques in different ambition levels per phase, which fits with the agile and modular improvements between cycles of this roadmap. The framework provides a concise context to the steps, ensuring that expectations and content are conveyed to the user. In addition, the relationship between usage, load, and condition (Tinga, 2010) is shown in the monitoring step. The required models and resulting accuracy are messages that contribute well to this step, as concluded in the second use case of this project.

The CRISP-DM model is used as a basis for the IT steps, with the more specified 'predictive maintenance for data scientists' (Sajid et al., 2021) steps next to it. The horizontal alignment between all frameworks shows the connection and similarities.

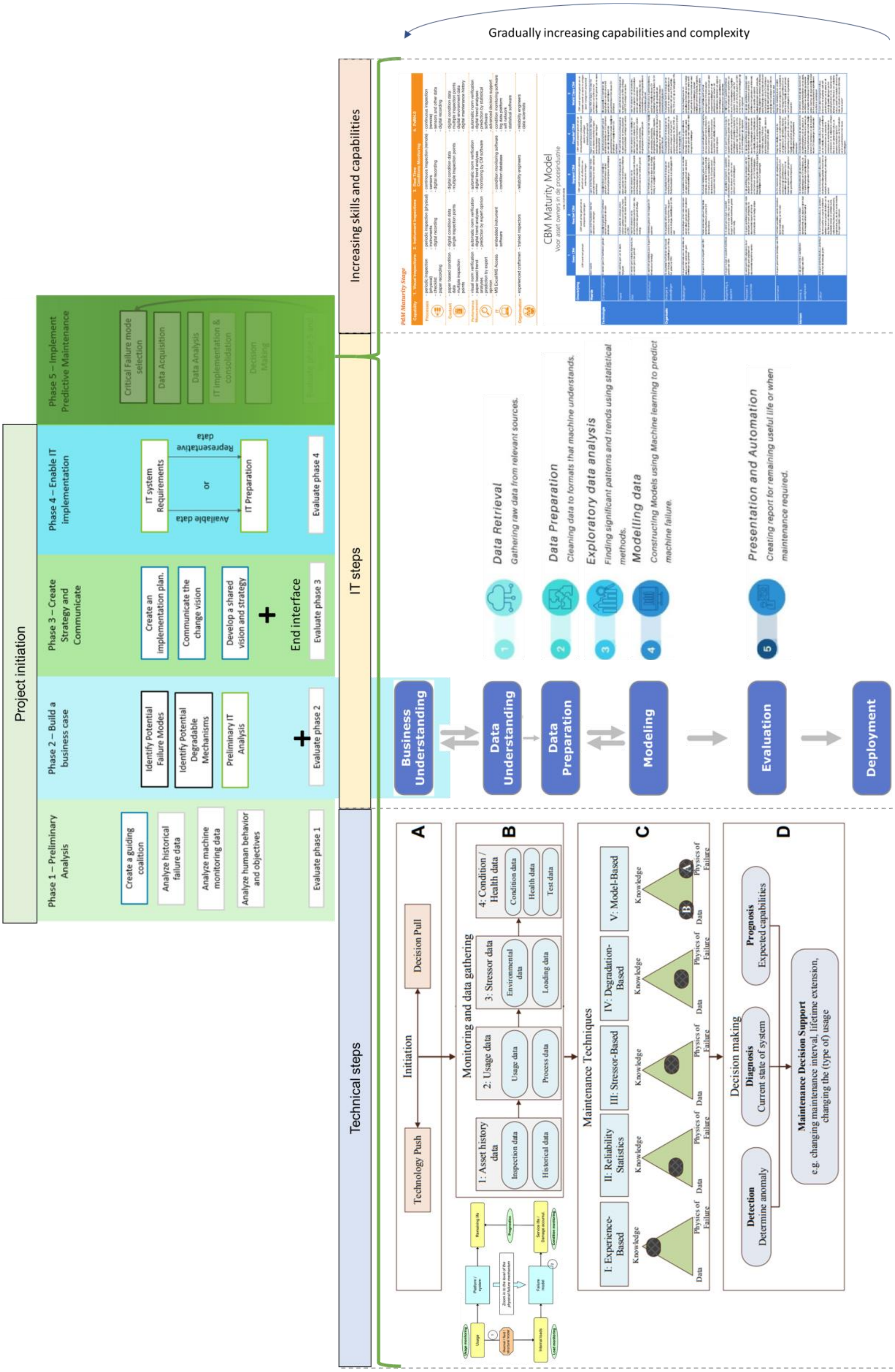


Figure 31, Roadmap agile and modular implementation data-driven maintenance

6.3. Implications

This section presents implications for Stork IMM, practical implications for other SMEs, and theoretical implications related to the literature view.

Stork IMM management implications

How can the data-driven maintenance application be further developed for Stork IMM, and how can this application play an essential role soon? To achieve this, the tool must effectively meet the business needs of Stork IMM's customers or itself. The final form must also be usable for customers. The business needs realized with a data-driven maintenance tool are minimizing downtime due to failures, minimizing production costs, and increasing production speed. The tool is most interesting to customers if it indicates failures early, can extend a failure until spare parts arrive, or provides direct suggestions on machine settings to improve operational properties. For Stork IMM, information about the use of machines can be logged with the data-driven maintenance tool. This information plays a major role in discussions in quality projects, developing functions, and developing structural improvements.

Many manufacturing companies have a manufacturing execution system (MES). So developing an application or dashboard to manage all machine activities exceeds its purpose. A customer would rather integrate data-driven conclusions into his MES of supply chain system. What would be best for Stork IMM is to develop simple applications that incorporate Stork IMM's knowledge about the machine. This information is unique and interesting for the customer. A tool can be realized in simple apps or integrated into the control system. This way, the customer does not have to deal with another dashboard to interpret, but using the data-driven suggestions is attractive and rewarding.

Practical implications

Industrial agile working is recognized as a good strategy in various companies that place themselves in higher maturity levels (Cimini et al., 2024). The findings in this research are comparable. We conclude that working agilely and modularly increases the success of implementing high-maturity, complex techniques. Modular implementation has led to several advantages:

- It is not an impossibly large project/hurdle, growing gradually
- Less dependent on advanced IoT/IT in the beginning
- Smaller steps promoting multidisciplinary collaboration
- Become more aware of the basic needs/skills required
- Being able to work towards goals in an adaptive manner
- Faster results, demonstrations, and value

In this research, the action research mindset led to the agile approach. Tailoring the modular steps to the company's current position is achievable by considering implementation frameworks, technology diagrams, maturity models, and the necessary IT infrastructure. As shown in the roadmap in Figure 31, this implementation process can be a good guide for SMEs.

For SMEs, it is good to start logging data. Logged data and information are the basis for developing data-driven applications. The lack of logged data can severely limit the progress and depth of conclusions.

Theoretical implications

Many technical obstacles in this project have already emerged in the literature review. Conducting this study agrees with many conclusions from other studies. For example, the loss of accuracy of a forecast due to more calculation models, which happened in the second use case, has been schematically mapped by (Tinga, 2010) in Figure 6. In addition, the forecast was unsuccessful in the first use case due to the remaining uncertainty that can be answered with historical data. This data is missing due to the discussed data consistency and availability problems. (Tiddens, 2018) has expressed the need for multiple data sources for our ambition level, as shown in Figure 10.

Furthermore, the steps to get where we are now can often be found in the existing frameworks. Looking at Primavera, for example, these exact steps have been taken to achieve data-driven maintenance. A challenge for an implementer is to consider all available written information. Reading for knowledge involves delving into various sources, each contributing unique insights to the literature review. The gap between theory and practice in applying data-driven maintenance makes it challenging to choose a suitable maintenance approach, identify appropriate components, and evaluate the required investments (Tiddens, 2018). The roadmap presented in Figure 31 helps implementers oversee and structure information for implementing data-driven maintenance.

Most existing frameworks mainly contain the technical steps required to implement data-driven maintenance. An exception, for example, is the CRISP-DM model, which indicates the importance of business understanding. In addition, (Van Eijk, 2023) draws attention to change management within often stiff companies. In contrast to Van Eijk's framework, which primarily aims to transform work processes to facilitate data-driven maintenance, this research integrates applications in parallel to existing processes. As OEM, data-driven maintenance is an extra service that can be provided. In the application within this company, it seems better to develop a tool that relieves the customer of a problem so that the tool is automatically used more.

A data-driven maintenance tool must address a business need and be embraced by personnel to generate value. This underscores a straightforward yet crucial finding: figuring out business needs and envisioning the final outcome early on is beneficial. In addition to Van Eijk's framework, it would be good to determine the end user's business needs and the form for the end user in the first phases of the framework. If the tool effectively fulfills a business need, it pays to use it more automatically. The end form must also be selected in the initial phases to develop the application. The final form could, for example, be an integration into the machine control system, a simple app, a dashboard, or a periodic email. The improvements in the framework have been included in the roadmap in Figure 31.

6.4. Action Research

In carrying out our research, action research provided a unique mindset. It helped us get to work proactive and perform activities gradually better. By doing this we achieved faster results compared to standard research, and this helped us adjust the project. Adjustment was necessary because we gradually discovered where the data-driven maintenance technique was valuable. It also helped to make certain things easier to try without the risk of it being the wrong technique. For example, we tried different things for the IoT infrastructure and experienced what it was like. As opposed to pre-determining what seems best, this turned out to be effective.

The learning lessons during the execution of the different cycles were sometimes not fully expressed. There were learning lessons such as demonstrating value with the dashboard and adding a productivity use case that emerged clearly. However, in basic activities like data handling, the learning curve is also steep. Knowledge about effective methods and programs gets automatically integrated into subsequent research cycles.

In this study, the second cycle has already started before the first cycle has been completed. It is logical that this happens due to waiting times and new emerging cases. In this study it was not a problem because the experience of the individual steps of the first cycle was used in the second cycle.

It was a challenge to concretely divide the work according to the specific action research steps. In the first cycle it takes time to properly understand the content and function of the steps. The line was sometimes thin between activities of steps. The line is also thin between reflecting on the content of the case and processes to realize the case. This was a very big challenge while reporting the study.

Action research has definitely resulted in changes. Use cases have been implemented and we have learned from these steps in the implementation process. The aim of this research was to learn from these first steps. If a study seeks conclusions deeper into the technology, action research can be an obstructive method because preparatory problems can take quite a lot of time. In this project, the action research method has led to the correct attitude, so its characteristic has been included in the Agile and Modular Implementation Roadmap.

6.5. Limitations

The research was implemented in one company. The focus and conclusions arose in this specific context. However, similar conclusions have been found in theory from implementation at other companies. Despite the generalization by comparison to other research conclusions, there may be slight tunnel vision or biases mixed in the conclusions.

Due to understaffing in the software department and time/benefit consideration with other projects, developing a new logging functionality for the third use case was not feasible in the projected amount of time. Although the third use case certainly shows the potential of using the same logging infrastructure for machine optimization, there is much potential for improvement. This conclusion also characterizes a common problem for SMEs, where limited personnel has to make the trade-off between projects.

Unfortunately, carrying out a forecast was unsuccessful due to missing historical data. The historical data that initially seemed available is missing due to data availability and consistency issues. The resulting logging system now monitors failures so that this reference data is built up. Due to the absence of this data, the result is not a prediction or prediction algorithm but a degradation visualization.

6.6. Further research

More profound research can be done to assist in creating work packages for modular implementation. Improving the roadmap from Figure 31 would help implementers oversee crucial information and messages from various researchers. How can companies gradually improve and how can the necessary information to tailor the implementation actions to the company's specific context be made understandable? The same applies to the literature on data-driven maintenance techniques. Much research has been done on condition monitoring and predictive maintenance. Overseeing the amount of information is a challenge. However, it has become apparent that the message can be essential in many pieces of research. Take, for example, the required data for a certain ambition level defined by (Tiddens, 2018), a detail that comes back as crucial in the end.

7. Conclusion

This project successfully implemented data-driven maintenance in SME Stork IMM. By applying action research as a research structure, the technique was implemented agilely and we were able to observe and improve the implementation process. The individual research questions are addressed in separate paragraphs.

How can data-driven analysis techniques be used to monitor the condition of Stork IMM injection molding machine components?

Data must be analyzed to apply data-driven maintenance techniques, typically involving a CRISP-DM data mining analysis. This framework helps with complex data mining issues and data science challenges such as data completeness, consistency, and availability. The analysis can be performed on edge, i.e., at the injection molding machine or at Stork IMM, the OEM. This connection is provided by an IoT infrastructure, realized in this project by Stork IMM itself for the best compatibility with the assets. Analyzing the sensor data with the necessary algorithms can result in a measured or calculated condition. The use cases have confirmed that the highest accuracy is achieved when the condition or governing load is measured directly. A condition in itself is often challenging to display, but the use cases have shown that displaying the load on a system and a degradation pattern can already achieve the requested objective.

How can data-driven maintenance be used to increase the reliability of Stork IMM machines?

In the application of data-driven maintenance, it is important to determine the intended ambition level. One must wonder whether a high ambition level is technically and economically feasible and what information is needed to decide on a maintenance action. With increasing complexity, possible outcomes can be descriptive, diagnostic, predictive, and prescriptive conclusions. This project has shown that descriptive and diagnostic results can be valuable in assessing a situation. These should be the first goals for the implementation of data-driven maintenance. Based on accumulated experience with this data, predictive and prescriptive conclusions can be the next step. To illustrate this, the tie-bar fracture has been diagnosed once with our monitoring analysis, but we still need to prove whether this pattern always has the same properties to make it fully predictable. From now on, experience is being built to recognize and predict this pattern in the logged data.

What organizational needs and requirements should be considered to implement data-driven maintenance effectively in Stork IMM's specific operational environment?

A gap is recognized between theory and practice in data-driven maintenance strategies. Several data-driven maintenance frameworks have been created to fill this gap, mainly focused on the technical steps. There are many frameworks and studies, each with a specific message. It is difficult for an implementer to oversee the amount of theory and consider every message. The Data-Driven Maintenance Techniques Framework (Tiddens, 2018) provides the best information about the technical application compared to the results of this research by incorporating different methods with increasing ambition levels in every step from start to result. The Proactive Maintenance Transformation Framework from (Van Eijk, 2023) highlights the preparatory steps for the IT system focussed on SMEs. Based on the feedback in this project, a good addition to this framework would be to determine the end user's business needs and the end user's final form in the preparatory phase. The Agile and Modular Implementation Roadmap includes the preparatory steps with improvements and the link between the technical and IT frameworks. The roadmap was created from a combination of the most valuable frameworks in this research. The roadmap should help future implementers use a practical approach to implementing data-driven maintenance.

This project was successful by modularly implementing data-driven maintenance. The agile properties of the action research methodology initiated this approach. By implementing data-driven maintenance modularly, awareness of the required skills in different areas could be built up, results and value were produced faster, the final form could be adjusted during the project, and it promoted interdisciplinary collaboration. An organization must understand the basic needs and skills to implement the technical frameworks. Maturity frameworks can help assess the current levels of skills on required capabilities. The Agile and Modular Implementation Roadmap has added a cyclical step to improve the capabilities, complexity, and automation in cycles. Ultimately, the project showed that most of the time was spent developing new functionalities on the IT or technical side. New use cases that use the same infrastructure and algorithms are more straightforward to add. Finally, the value created with the system must fulfill a business need and correspond to the intended ambition level to be used in daily activities.

Wrapping up: How to implement and leverage data-driven maintenance in SME Stork IMM?

Implementing and utilizing data-based information for maintenance decisions did previously not take off at Stork IMM, a problem recognized generically in SMEs. We applied data-based maintenance at Stork IMM through iterative action research cycles using literature on the technology, technical frameworks, and implementation frameworks. Several hurdles have emerged during the implementation process, with many realizations about the lack of skills and system capabilities typical of an SME. We concluded that if data-driven maintenance can fulfill a business need for an SME, it can be implemented effectively by implementing the frameworks and techniques in a modular manner. This approach enables the gradual development of complexity and required skills for the desired ambition level, leading agilely to valuable results.

8. References

Badea, V. E., Zamfiroiu, A., & Boncea, R. (2018). Big Data in the Aerospace Industry. *Informatica Economica*, *22*(1/2018), 17–24. https://doi.org/10.12948/issn14531305/22.1.2018.02

Cachada, A., Barbosa, J., Leitao, P., Geraldes, C., & Deusdado, L. (2018). *Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture*.

Chapman, P., Clinton, J., Kerber, R., Khabaz, T., & Reinartz, T. (2000). *CRISP-DM 1.0; Step-by-step data mining guide*. DaimlerChrysler.

Cimini, C., Lagorio, A., & Cavalieri, S. (2024). Development and application of a maturity model for Industrial Agile Working. *Computers & Industrial Engineering*, *188*, 109877. https://doi.org/10.1016/j.cie.2023.109877

Cohen, L., Manion, L., & Morrison, K. (2017). *research Methods in Education* (8th ed.). Routledge.

Coughlan, P., & Coghlan, D. (2002). Action research for operations management. *International Journal of Operations and Production Management*, *22*(2), 220–240. https://doi.org/10.1108/01443570210417515

Data Governance Act. (2022). *REGULATION (EU) 2022/868 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act)*.

Decker, P., Durand, R., Mayfield, C. O., McCormack, C., Skinner, D., & Perdue, G. (2012). Predicting implementation failure in organization change. *Journal of Organizational Culture, Communications and Conflict*, *16*(2), 29–30.

Druetta, C. (2018). Legal Perspectives on Predictive Maintenance: A Case Study; Legal Counsel, Comau. *International In-House Counsel Journal*, *11*(44). https://www.plattform-i40.de/I40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html.

European Commission. (2023). *Internal Market, Industry, Entrepreneurship and SMEs; SME definition*. https://single-market-economy.ec.europa.eu/smes/sme-definition_en

George, T. (2023). *What Is Action Research? | Definition & Examples.* https://www.scribbr.com/methodology/action-research/

Gummesson, E. (2000). *Qualitative Research Methods in Management Research* (Vol. 44). Journal of The Operational Research Society.

Humbeck, P., Mangold, S., & Bauernhansl, T. (2020). Future scenarios of value creation in mechanical engineering - Derivation of recommendations for action. *Procedia CIRP*, *93*, 844–849. https://doi.org/10.1016/j.procir.2020.03.093

Ingemarsdotter, E., Kambanou, M. L., Jamsin, E., Sakao, T., & Balkenende, R. (2021). Challenges and solutions in condition-based maintenance implementation - A multiple case study. *Journal of Cleaner Production*, *296*. https://doi.org/10.1016/j.jclepro.2021.126420

Kerkhof, R. van de. (2020). *Het implementeren en opschalen van Condition-Based Maintenance LESSONS LEARNED Het implementeren en opschalen van Condition-Based Maintenance*. 37.

Krishna Durbhaka, G., & Selvaraj, B. (2021). Convergence Of Artificial Intelligence And Internet Of Things In Predictive Maintenance Systems-A Review. In *Turkish Journal of Computer and Mathematics Education* (Vol. 12, Issue 11).

Lee, J., De Pater, I., Boekweit, S., & Mitici, M. (2022). *Remaining-Useful-Life prognostics for opportunistic grouping of maintenance of landing gear brakes for a fleet of aircraft*.

Leedy, P. D., & Ormrod, J. E. (2015). *Practical Research; Planning and Design* (Eleventh edition). Pearson Global Edition.

Mainnovation. (2018). *Predictive maintenance 4.0; Beyond the hype: PdM 4.0 delivers results*.

Matt, D. T., Modrák, V., & Zsifkovits, H. (2020). Industry 4.0 for smes: Challenges, opportunities and requirements. In *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*. Palgrave Macmillan. https://doi.org/10.1007/978-3-030-25425-4

Mesarosova, J., Martinovicova, K., Fidlerova, H., Chovanova, H. H., Babcanova, D., & Samakova, J. (2022). IMPROVING THE LEVEL OF PREDICTIVE MAINTENANCE MATURITY MATRIX IN INDUSTRIAL ENTERPRISE. *Acta Logistica*, *9*(2), 183–193. https://doi.org/10.22306/al.v9i2.292

Mitici, M., de Pater, I., Barros, A., & Zeng, Z. (2023). Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics: The case of turbofan engines. *Reliability Engineering and System Safety*, *234*. https://doi.org/10.1016/j.ress.2023.109199

NEN-EN. (2019). *NEN-EN 13306*. www.nen.nl

Sajid, S., Haleem, A., Bahl, S., Javaid, M., Goyal, T., & Mittal, M. (2021). Data science applications for predictive maintenance and materials science in context to Industry 4.0. *Materials Today: Proceedings*, *45*, 4898–4905. https://doi.org/10.1016/j.matpr.2021.01.357

Sensorfy B.V. (2023). *The essential guide to predictive maintenance for OEMs*.

Shapiro, J. (2023, January 26). Data Driven at 200 MPH: How Analytics Transforms Formula One Racing. *Forbes*.

Spendla, L., Kebisek, M., Tanuska, P., & Hrcka, L. (2017). Concept of predictive maintenance of production systems in accordance with industry 4.0. *SAMI 2017 - IEEE 15th International Symposium on Applied Machine Intelligence and Informatics, Proceedings*, 405–410. https://doi.org/10.1109/SAMI.2017.7880343

Sullivan, G. P., Pugh, R., Melendez, A. P., & Hunt, W. D. (2010). *Operations & Maintenance Best Practices A Guide to Achieving Operational Efficiency*.

Tiddens, W. W. (2018). *Setting sail towards predictive maintenance : developing tools to conquer difficulties in the implementation of maintenance analytics* [University of Twente]. https://doi.org/10.3990/1.9789036546034

Tiddens, W. W., Braaksma, A. J. J., & Tinga, T. (2015). The Adoption of Prognostic Technologies in Maintenance Decision Making: A Multiple Case Study. *Procedia CIRP*, *38*, 171–176. https://doi.org/10.1016/j.procir.2015.08.028

Tiddens, W. W., Braaksma, J., & Tinga, T. (2020). Exploring predictive maintenance applications in industry. *Journal of Quality in Maintenance Engineering*, *28*(1), 68–85. https://doi.org/10.1108/JQME-05-2020-0029

Tiddens, W. W., Braaksma, J., & Tinga, T. (2023). Decision Framework for Predictive Maintenance Method Selection. *Applied Sciences (Switzerland)*, *13*(3). https://doi.org/10.3390/app13032021

Tinga, T. (2010). Application of physical failure models to enable usage and load based maintenance. *Reliability Engineering and System Safety*, *95*(10), 1061–1075. https://doi.org/10.1016/j.ress.2010.04.015

Tinga, T. (2012). *MechanisM based failure analysis*.

Ton, B., Basten, R., Bolte, J., Braaksma, J., Di Bucchianico, A., van de Calseyde, P., Grooteman, F., Heskes, T., Jansen, N., Teeuw, W., Tinga, T., & Stoelinga, M. (2020). Primavera: Synergising predictive maintenance. *Applied Sciences (Switzerland)*, *10*(23), 1–19. https://doi.org/10.3390/app10238348

Van de Kerkhof, R. (2020). *It's about time: Managing implementation dynamics of condition-based maintenance*. CentER, Center for Economic Research.

Van Eijk, B. (2023). *Bridging the Gap in Maintenance: A Practical Framework for Implementing Predictive Maintenance in Small and Medium Enterprises*.

Wang, K. (2016). *Intelligent Predictive Maintenance (IPdM) system-Industry 4.0 scenario*. https://doi.org/10.2495/IWAMA150301

Yokogawa. (2020). *Digital Transformation in Process Industries: A Journey Towards Autonomous Operations*.

Appendix A. Usage of AI tools

With the increasing potential of AI tools, a declaration of the use of AI has been added to this appendix, as recommended by the University of Twente.

During the preparation of this work the author used ChatGPT in order to translate and improve the required to program functionalities in the programming language Python. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

Furthermore, Mendeley is used as a reference manager, and Grammarly is used as a spelling checker assistant.

Appendix B. Maturity models

Maturity model from (Van de Kerkhof, 2020):

CBM Maturity Model

Voor asset owners in de procesindustrie

versie 2.3 (25-03-2020)

		1 Geen CBM	2 Reactief CBM	3 Gepland CBM	4 Proactief CBM	5 World Class CBM
	Omschrijving	*CBM wordt niet gebruikt*	*CBM wordt ad hoc gebruikt om te anticiperen op storingen*	*CBM wordt structureel en planmatig gebruikt om de efficiëntie van onderhoud te verhogen*	*CBM wordt proactief gebruikt om de betrouwbaarheid en productiviteit van assets te verhogen (reliability perspectief)*	*CBM wordt optimaal gebruikt om de behaalde waarde uit assets te verhogen (asset management perspectief)*
	Waarde	Geen waarde	Lagere onderhoudskosten door het voorkomen van storingen	Lagere onderhoudskosten door minder correctief onderhoud en beter voorbereid onderhoud	Hogere omzet uit productie door hogere OEE en klanttevredenheid en lagere onderhoudskosten door hogere betrouwbaarheid	Hogere ROA en lagere TCO door het verminderen van buffers en het optimaliseren van het gebruik van de assets
Technologie	CM technologieën	Er worden geen CM technieken gebruikt	Makkelijk-te-gebruiken CM technieken worden enkel gebruikt als nader onderzoek	Makkelijk-te-leren en makkelijk-te-gebruiken CM technieken worden structureel gebruikt (proven technologies)	Er wordt structureel onderzocht wat de optimale (combinatie) van CM technieken per asset zijn. Hierbij wordt ook geëxperimenteerd met enkele moeilijk-te-ontwikkelen en moeilijk-te-leren CM technieken	Alle succesvolle CM technieken zijn opgeschaald en worden structureel gebruikt. Er wordt blijvend geëxperimenteerd met nieuwe CM technieken
	Assets	CBM wordt op geen van de assets toegepast	Enkel de assets die vanwege andere redenen geobserveerd worden, maken kans op een verzoek tot nader onderzoek en CBM	CBM wordt structureel toegepast op de assets waarvoor het onderhoud efficiënter uitgevoerd kan worden	CBM wordt ook structureel toegepast op de assets waarvoor de betrouwbaarheid en/of productiviteit verhoogd kan worden	CBM wordt ook structureel toegepast op de assets waarvoor de ROA verhoogd en/of TCO verlaagd kan worden
	Data	Er worden geen analyses uitgevoerd, dus er worden geen data gebruikt	Voor het uitvoeren van de onderhoudsanalyses worden master data en instrument data gebruikt (huidige meting)	Voor het uitvoeren van de onderhoudsanalyses worden ook financiële onderhoudsdata en inspectie- en instrument data uit het verleden gebruikt	Voor het uitvoeren van de reliability- en risico management analyses en het ontwikkelen van CM technieken worden ook procesdata, productdata, omgevingsdata en faaldata gebruikt	Voor het uitvoeren van de productie-, inkoop-, projecten- en ontwerpanalyses worden ook voorraaddata en voorspellingen/toekomstige data van productieplanning, omgevingscondities en marktcondities gebruikt
	IT-infrastructuur	Er wordt niet gemonitord, dus er is geen IT-infrastructuur benodigd	Monitoring gebeurt met draagbare CM systemen	De IT-infrastructuur maakt het ook mogelijk om de CM data op te slaan en de huidige meting te vergelijken met historische data	De IT-infrastructuur maakt het ook mogelijk om procesdata, productdata, omgevingsdata en faaldata te koppelen, zowel voor het ontwikkelen van nieuwe CM toepassingen en voor het structureel gebruiken hiervan	De IT-infrastructuur is gestandaardiseerd, zodat het makkelijk is nieuwe CM toepassingen hierop aan te sluiten. De CM systemen zijn gekoppeld aan productieplannings-, inkoop en procesaansturingssystemen
Organisatie	Strategie & doelstellingen	De organisatie heeft (al dan niet bewust) geen strategie, doelstellingen en KPIs op het gebied van CBM	De organisatie wil het onderhoud verbeteren, maar heeft hier nog geen concrete strategie, doelstellingen en KPIs voor	De organisatie heeft de strategie om onderhoud efficiënter uit te voeren. Onderhoudskosten is de belangrijkste KPI	De organisatie heeft de strategie om de betrouwbaarheid en productiviteit van de assets te verhogen en heeft een CM programma opgestart. De OEE, MTBF en onderhoudskosten/geproduceerd product zijn de belangrijkste KPIs	De organisatie heeft de strategie om de waarde uit de assets te optimaliseren en commiteert zich aan een CM portfolio. ROA, TCO en LCC zijn de belangrijkste KPIs
	Beslissingen	Er is geen informatie over de conditie van assets, dus hier worden ook geen beslissingen op genomen	De bevindingen uit het nader onderzoek worden alleen gebruikt voor het plannen van het onderhoudsmoment	De periodieke informatie over de conditie van assets wordt gebruikt voor (meer) onderhoudsbeslissingen	De hoogfrequente en gedetailleerde informatie over de conditie van assets wordt ook gebruikt voor reliability- en risico management beslissingen	De brede, hoogfrequente en gedetailleerde informatie over de huidige en toekomstige conditie van assets wordt ook gebruikt in een breed scala aan asset management beslissingen, inclusief beslissingen omtrent productie, projecten, inkoop en ontwerp van (nieuwe) assets
	Structuur	Er is geen structuur ingericht voor CBM	Nader onderzoek gebeurt door lokale onderhoudsteams en externe CM dienstverleners	Structurele monitoring gebeurt door een combinatie van lokale CM teams, centrale CM teams en externe specialistische CM dienstverleners. De CM teams werken nauw samen met maintenance engineers	Er is een centraal ingericht CM programma, dat nauw samenwerkt met de interne CM teams en externe specialistische CM dienstverleners. De CM teams werken nauw samen met reliability engineers en process engineers	Het CM portfolio wordt centraal gemanaged. De CM teams worden intensief betrokken bij een reeks aan asset management beslissingen en zijn geïntegreerd in een netwerk van kennisinstituten, fabrikanten van assets en CM technologieën, specialistische CM dienstverleners en data scientists
	Budgettering & capaciteit	Er is geen budget & capaciteit beschikbaar gesteld voor CBM	Er is vooraf geen budget & capaciteit gereserveerd voor CBM, maar er wordt wel budget & capaciteit beschikbaar gesteld wanneer nodig	Er zijn jaarlijkse budgetten & capaciteiten beschikbaar gesteld voor het uitvoeren van CM, het uitvoeren van CBM en het beheren van CM technologieën	Er is een apart CM programma budget & capaciteit beschikbaar gesteld voor het ontwikkelen en aanschaffen van nieuwe CM technologieën. De jaarlijkse budgetten & capaciteiten voor het uitvoeren van CM, het uitvoeren van CBM en het beheren van CM technologieën zijn uitgebreid	Er blijft budget & capaciteit beschikbaar voor het ontwikkelen en aanschaffen van nieuwe CM technologieën. De jaarlijkse budgetten & capaciteiten voor het uitvoeren van CM, het uitvoeren van CBM en het beheren van CM technologieën zijn verder uitgebreid
	Processen & documentatie	Er wordt geen CBM uitgevoerd, dus er hoeven ook geen processen en documentatie ingericht te worden	Er is geen gedefinieerd proces voor nader onderzoek en werk uit inspectie. De documentatie beperkt zich tot de communicatie van de huidige analyse	Er zijn gedefinieerde processen voor het uitvoeren van CBM, die geïntegreerd zijn in de standaard onderhoudswerkprocessen, en het beheren van CM technologieën. Belangrijke documentatie omvat (standaard) inspectielijsten en CM rapportages	Er zijn ook gedefinieerde processen voor het ontwikkelen en implementeren van nieuwe CM toepassingen, het uitvoeren van reliability analyses en modificaties en het evalueren van onderhouds-concepten. Belangrijke documentatie omvat een lijst met kritische assets, FMEAs en onderhoudsconcepten van die assets en CM concepten uit de pilots	Er zijn ook gedefinieerde processen voor het continu verbeteren van het CM portfolio en het gebruiken van informatie over de conditie van assets in beslissingsprocessen omtrent productie, inkoop, projecten en ontwerp van (nieuwe) assets. Belangrijke documentatie omvat een actueel overzicht van de CM technieken die bij elk asset gebruikt worden, een actuele lijst met kandidaten voor CBM en een CM concept per type asset
	Governance	Er is geen governance benodigd voor CBM	Technisch specialisten worden betrokken bij de beoordeling van het nader onderzoek	De CM momenten zijn vastgelegd in een onderhoudsmanagementsysteem, CM procedures zijn gedefinieerd, CM specialisten zijn gecertificeerd en de inspectierapporten worden goedgekeurd door gecertificeerde inspecteurs	Design for reliability en design for maintenance zijn een verplicht onderdeel van projecten, er zijn heldere afspraken met interne en externe partijen over het eigenaarschap en gebruik van data en waar mogelijk wordt gebruik gemaakt van technologische en organisatorische standaarden	Design for monitoring is een verplicht onderdeel van projecten, de organisatie is asset management gecertificeerd en er wordt zo veel mogelijk gebruik gemaakt van technologische en organisatorische standaarden
Mensen	Kennis & vaardigheden	Er zijn geen kennis & vaardigheden benodigd voor CBM	De onderhoudsteams hebben domeinkennis van de assets en zijn in staat om te bepalen of iets 'normaal' is	De onderhoudsteams zijn ook bekend met de basisprincipes van CM technieken, de CM teams beheersen makkelijk-te-leren en makkelijk-te-gebruiken CM technieken	De onderhoudsteams zijn ook bekend met de faalmechanismen van de assets en in staat om FMEAs en RCAs uit te voeren, de CM teams beheersen ook enkele moeilijk-te-leren CM technieken en zijn in staat om nieuwe CM toepassingen te ontwikkelen	De betrokken teams zijn ook bekend met de dragradatiemechanismen van de assets, de effecten van degradatie op productie en de laatste innovaties op CM technologie gebied. Daarmee zijn zij gezamenlijk in staat om het productieproces te optimaliseren en buffers te verminderen
	Cultuur	Er is geen onderhoudscultuur, onderhoud wordt niet als belangrijk gezien	Er is een brandweercultuur, de personen die onverwachte en urgente problemen oplossen worden gezien als helden van de dag. Ook is er een eilandencultuur, de organisatie bestaat uit veel losse teams, zoals onderhoudsteams, productieteams, projectteams, etc., die elk in eerste instantie hun eigen doelen nastreven	Er is een bureaucratische cultuur, binnen de (onderhouds)organisatie heerst sterk de behoefte om procesmatig en planmatig te werken	Er is een reliability cultuur, het verhogen van de reliability wordt vanuit verschillende teams omarmd om de productie te verbeteren, de onderhoudskosten te verminderen en de veiligheid te verhogen. Ook is er een pionierende cultuur, de personen betrokken bij het CM programma houden van het ontwikkelen van en experimenteren met nieuwe technologieën	Er is een asset management cultuur, iedereen in de organisatie voelt zich gedeeld eigenaar van de assets en wil vanuit zijn positie bijdragen aan het optimaal gebruiken ervan, zowel op korte als lange termijn. Ook is er een analytische cultuur, waarin men besluiten wil nemen op basis van actuele en accurate informatie, "meten is weten"

Voor vragen over het CBM Maturity Model of het uitvoeren van een CBM Maturity Assessment, kun u contact opnemen met:

Roland van de Kerkhof
r.m.vdkerkhof@uvt.nl

Het CBM Maturity Model is mede ontwikkeld door: Nico van Kessel (Tata Steel), Vijay Mohan (BP), Niels Noorderhaven (UvT) & Henk Akkermans (UvT, WCM)

Afkortingen
CBM: Condition-Based Maintenance
CM: Condition Monitoring
OEE: Overall Equipment Effectiveness
ROA: Return On Assets
TCO: Total Cost of Ownership
LCC: Life Cycle Costs

Maturity model from (Mainnovation, 2018):

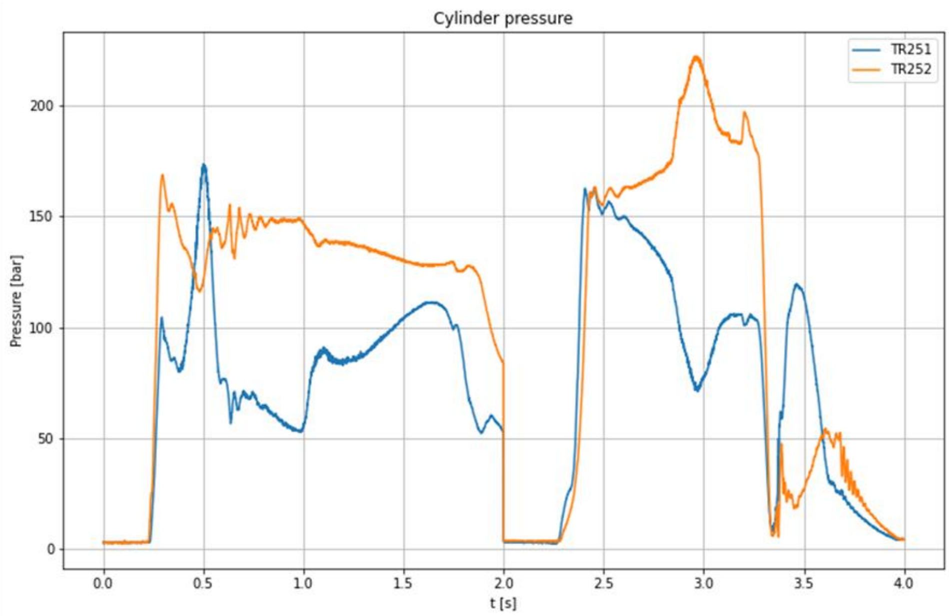
PdM Maturity Stage

Capability	1. Visual Inspections	2. Instrument Inspections	3. Real Time Conditions Monitoring	4. PdM4.0
Processes	- periodic inspection (physical) - checklist - paper recording	- periodic inspection (physical) - instruments - digital recording	- continouos inspection (remote) - sensors - digital recording	- continuous inspection (remote) - sensors and other data - digital recording
Content	- paper based condition data - multiple inspection points	- digital condition data - single inspection points	- digital condition data - multiple inspection points	- digital condition data - multiple inspection points - digital environment data - digital maintenance history
Performance Measurement	- visual norm verification - paper based trend analyses - prediction by expert opinion	- automatic norm verification - digital trend analyses - prediction by expert opinion	- automatic norm verification - digital trend analyses - monitoring by CM software	- automatic norm verification - digital trend analyses - prediction by statistical software - advanced decision support
IT	- MS Excel/MS Access	- embedded instrument software	- condition monitoring software - condition database	- condition monitoring software - big data platform - wifi network - statistical software
Organisation	- experienced craftsmen	- trained inspectors	- reliability engineers	- reliability engineers - data scientists

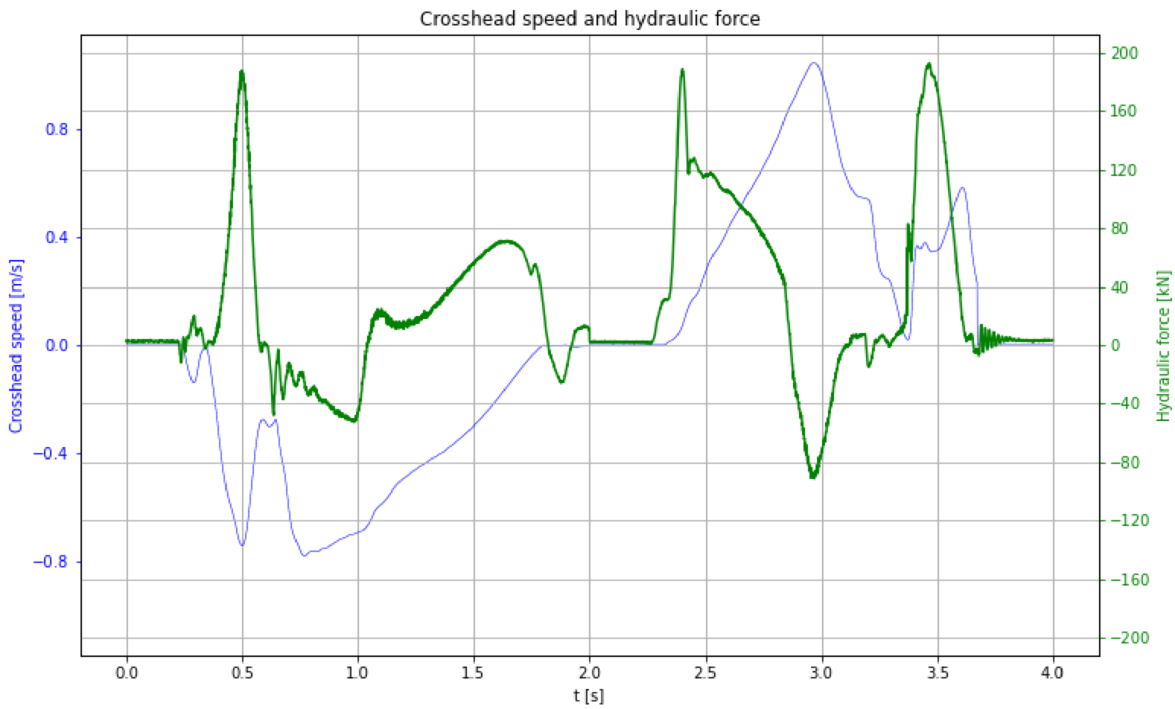
Appendix C. Monitoring frame load

The frameload is monitored using indirect sensors. This appendix gives more context to the data mining difficulties of a single cycle.

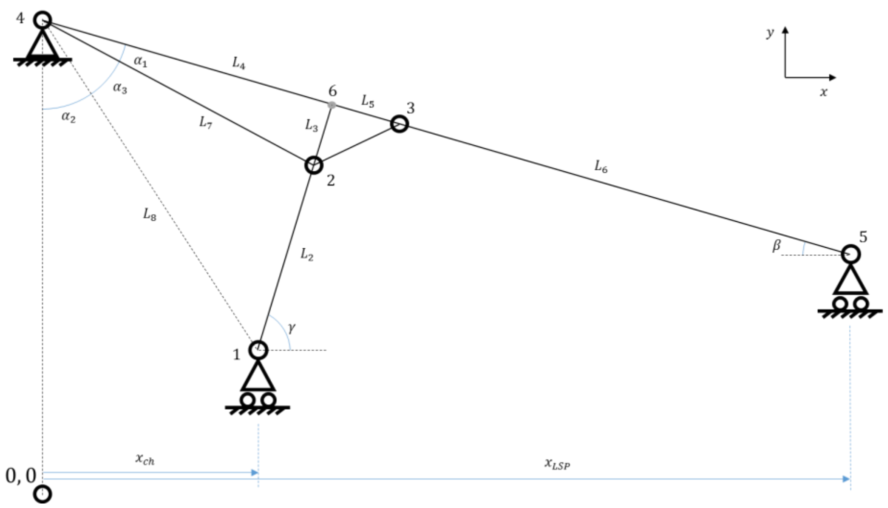
Measures cylinder pressures (first graph), and drive force (second graph):



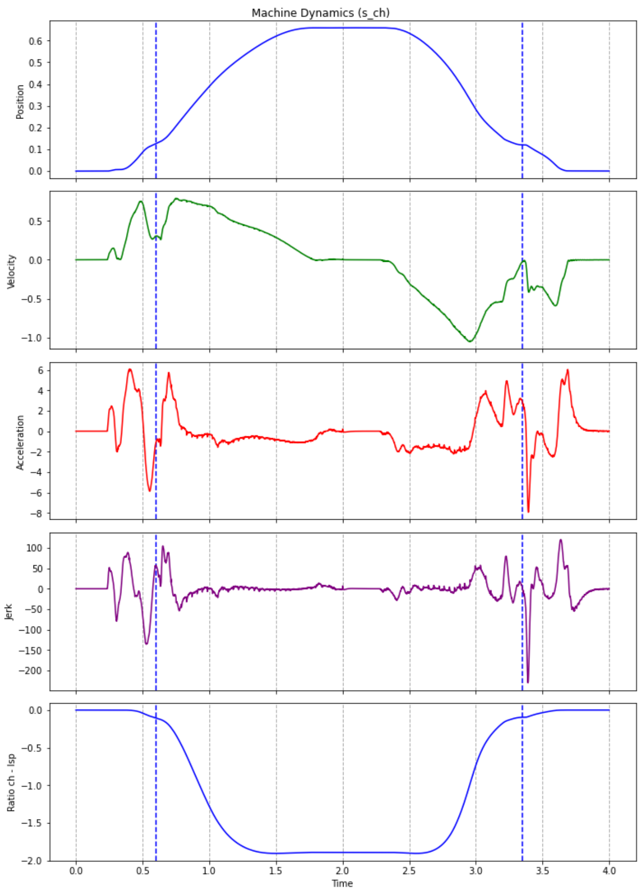
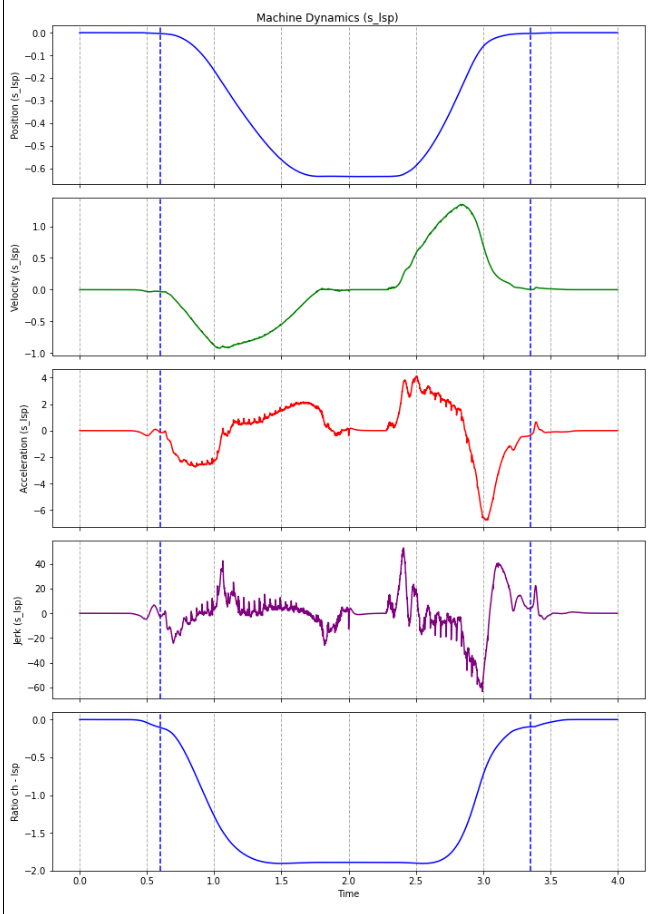
$$F_c = \frac{\pi}{4} * (p_p * d_p^2 - p_r * (d_p^2 - d_r^2))$$



Transmission system for LSP position algorithm:



Sensors and analysis to measure forces as a result of LSP acceleration:

Measured crosshead position, calculated velocity, acceleration and jerk	Calculated LSP position with transmission ratio, calculated velocity, acceleration and jerk
	

Appendix D. Code algorithms

Function 1: saving data from Stork DB to analysis DB

```python
import os
import gzip
import ctypes
import shutil
from datetime import datetime, timedelta

# This code transforms loggings from machine DB to logging DB

# Enter the machine number, start date and end date
machine_number = '81911'
start_date = datetime(2023, 11, 24)
end_date = datetime(2023, 12, 21)

def process_and_decompress_logs(machine_number, start_date, end_date):
    current_date = start_date
    while current_date <= end_date:
        date_str = current_date.strftime('%Y_%m_%d')

        input_directory = '\\\\l0\mach'
        output_directory = 'C:\\database_loggings'

        # Processing log_cycle file
        input_file_cycle = os.path.join(input_directory, machine_number, '.srs', f'log_cycle_{machine_number}_{date_str}.log.gz')
        destination_directory_cycle = os.path.join(output_directory, machine_number, 'loggings_txt')

        try:
            def toggle_hidden_attribute(path):
                try:
                    attrs = ctypes.windll.kernel32.GetFileAttributesW(path)
                    if attrs == -1:
                        return False
                    attrs ^= 2  # Toggle the hidden attribute (bit 1)
                    ctypes.windll.kernel32.SetFileAttributesW(path, attrs)
                    return True
                except Exception as e:
                    print(f"Error toggling hidden attribute: {e}")
                    return False

            toggle_hidden_attribute(os.path.dirname(input_file_cycle))

            if not os.path.exists(destination_directory_cycle):
                os.makedirs(destination_directory_cycle)

            toggle_hidden_attribute(os.path.dirname(input_file_cycle))

            output_file_cycle = os.path.join(destination_directory_cycle, f'log_cycle_{machine_number}_{date_str}.log')

            with gzip.open(input_file_cycle, 'rb') as f_in, open(output_file_cycle, 'wb') as f_out:
                f_out.writelines(f_in)

            print(f'{input_file_cycle} has been decompressed to {output_file_cycle}')

        except FileNotFoundError:
            print(f'File not found for date {date_str}. Skipping...')

        # Processing log_alarm file
        input_file_alarm = os.path.join(input_directory, machine_number, '.srs', f'log_alarm_{machine_number}_{date_str}.log')
        destination_directory_alarm = os.path.join(output_directory, machine_number, 'loggings_txt')

        try:
            toggle_hidden_attribute(os.path.dirname(input_file_alarm))

            if not os.path.exists(destination_directory_alarm):
                os.makedirs(destination_directory_alarm)

            toggle_hidden_attribute(os.path.dirname(input_file_alarm))

            output_file_alarm = os.path.join(destination_directory_alarm, f'log_alarm_{machine_number}_{date_str}.log')

            # Copy the log_alarm file to the destination directory
            shutil.copy2(input_file_alarm, output_file_alarm)

            print(f'{input_file_alarm} has been copied to {output_file_alarm}')

        except FileNotFoundError:
            print(f'File not found for date {date_str}. Skipping...')

        current_date += timedelta(days=1)

process_and_decompress_logs(machine_number, start_date, end_date)
```

Function 2: Data cleaning and data mining

```python
import os
import csv

machine_number = '81911'

tiebar_sensors = ['FOtot', 'FO1', 'FO2', 'FO3', 'FO4']
closeunit_sensors = ['ACclsM','PAclsM','ACclsm','PAclsm',\
                     'FOclsM','PFclsM','FOclsm','PFclsm',\
                     'ACopnM','PAopnM','ACopnm','PAopnm',\
                     'FOopnM','PFopnM','FOopnm','PFopnm']
sensors_of_interest = tiebar_sensors + closeunit_sensors
cycle_data_columns = ['UTC-time', 'Msec', 'Cycle-cnt']

directory = r'C:\database_loggings'
folder_path = os.path.join(directory, machine_number)

input_directory = os.path.join(folder_path, 'loggings_txt')

def process_logs_to_csv(machine_number, folder_path, sensors_of_interest, cycle_data_columns):
    # Initialize dictionaries to store data
    cycle_data = {col: [] for col in cycle_data_columns}
    sensor_data = {sensor: [] for sensor in sensors_of_interest}
    # Column count to seperate cycle data from sensor data
    column_count = 0
    # Set to store encountered cycle numbers
    encountered_cycle_numbers = set()

    # Iterate through the log files in the specified folder
    for filename in os.listdir(input_directory):
        if filename.startswith(f'log_cycle_{machine_number}_'):
            file_path = os.path.join(input_directory, filename)

            # Open the log file for reading
            with open(file_path, 'r') as file:
                lines = file.readlines()

                # Count column names if not done yet
                if not column_count:
                    column_names = lines[0].strip().split(',')
                    column_count = len(column_names)

                # run through lines of log file
                for line in lines[1:]:
                    values = line.strip().split(',')

                    # Check if the cycle number is already encountered
                    cycle_number = values[column_names.index('Cycle-cnt')]
                    if cycle_number in encountered_cycle_numbers:
                        # Skip this line
                        continue

                    # Add the cycle number to the set
                    encountered_cycle_numbers.add(cycle_number)

                    if len(values) > column_count:
                        # Extract cycle data into cycle data dictionary
                        for col in cycle_data_columns:
                            cycle_data[col].append(values[column_names.index(col)])

                        # Extract sensor data and split by sensor
                        sensor_data_str = ",".join(values[column_count:])
                        sensor_readings = sensor_data_str.split(';')

                        # If sensor is of interest, store reading in dictionary
                        for sensor_reading in sensor_readings:
                            parts = sensor_reading.split(',')
                            if len(parts) == 2 and parts[0] in sensors_of_interest:
                                sensor_name, sensor_value = parts[0], parts[1]
                                sensor_data[sensor_name].append(sensor_value)


    # Make CSV file and open to write into it
    csv_file_path = os.path.join(folder_path, f'loggings_{machine_number}.csv')
    with open(csv_file_path, 'w', newline='') as csv_file:
        writer = csv.writer(csv_file)

        # Column names
        csv_columns = cycle_data_columns + sensors_of_interest
        writer.writerow(csv_columns)

        # Determine amount of rows and write data from dictionaries to CSV file
        num_rows = min(len(cycle_data[cycle_data_columns[0]]), len(sensor_data[sensors_of_interest[0]]))
        for i in range(num_rows):
            row = [cycle_data[col][i] for col in cycle_data_columns] + [sensor_data[sensor][i] for sensor in sensors_of_interest]
            writer.writerow(row)
```

```python

process_logs_to_csv(machine_number, folder_path, sensors_of_interest, cycle_data_columns)

output_csv_path = os.path.join(folder_path, f'loggings_alarms_{machine_number}.csv')

# Write the header once outside the loop
header = ['Type', 'UTC-time', 'Msec', 'Cycle-cnt', 'Sub-index', 'AlarmIndex', 'Status', 'Context']

with open(output_csv_path, 'w', newline='') as output_csv:
    csv_writer = csv.writer(output_csv)
    csv_writer.writerow(header)

    # Iterate through files in the input directory
    for file_name in os.listdir(input_directory):
        # Check if the file starts with 'log_alarm_'
        if file_name.startswith('log_alarm_'):
            file_path = os.path.join(input_directory, file_name)

            # Read the file content
            with open(file_path, 'r') as file:
                lines = file.readlines()

            # Append the rows to the output CSV file
            csv_writer.writerows([line.strip().split(',') for line in lines[1:]])

```

Function 3: Data analysis

```python
import os
import pandas as pd
import numpy as np

# Initialization:
machine_number = '81911'
closing_force = 11000     #closing force in kN
analyzed_cycles = 1000  #amount of cycles to analyse

# Import loggings:
directory = r'C:\database_loggings'
csv_input = f'loggings_{machine_number}.csv'
csv_input_alarms = f'loggings_alarms_{machine_number}.csv'
input_file = os.path.join(directory, machine_number, csv_input)
input_file_alarms = os.path.join(directory, machine_number, csv_input_alarms)
machinedata = pd.read_csv(input_file)
loggings_alarms = pd.read_csv(input_file_alarms)

# Output files:
csv_analysis_calculations = os.path.join(directory, machine_number, 'analysis_calculations.csv')
csv_analysis_results = os.path.join(directory, machine_number, 'analysis_results.csv')
csv_analysis_accelerations = os.path.join(directory, machine_number, 'analysis_accelerations.csv')
csv_overloads_tiebars = os.path.join(directory, machine_number, 'overloads_tiebars.csv')
csv_overloads_frame = os.path.join(directory, machine_number, 'overloads_frame.csv')
csv_production_KPIs = os.path.join(directory, machine_number, 'production_KPIs.csv')
csv_production_alarms = os.path.join(directory, machine_number, 'production_alarms.csv')

# last cycle:
cycle_count = machinedata['Cycle-cnt'].iloc[-1]

# Convert date column
machinedata['UTC-time'] = pd.to_datetime(machinedata['UTC-time'],unit='s')
loggings_alarms['UTC-time'] = pd.to_datetime(loggings_alarms['UTC-time'],unit='s')

# Time difference to London
machinedata['UTC-time'] = machinedata['UTC-time'] + pd.Timedelta(hours=1)
loggings_alarms['UTC-time'] = loggings_alarms['UTC-time'] + pd.Timedelta(hours=1)

last_log_date = machinedata['UTC-time'].max().date()

#columns
cycle_data_columns = ['UTC-time', 'Msec', 'Cycle-cnt']
tiebars = ['F01', 'F02', 'F03', 'F04']
Force_sensors = ['FOclsM','FOclsm','FOopnM','FOopnm']
Acceleration_sensors = ['ACclsM','ACclsm','ACopnM','ACopnm']
closeunit_sensors = ['ACclsM','PAclsM','ACclsm','PAclsm',\
                     'FOclsM','PFclsM','FOclsm','PFclsm',\
                     'ACopnM','PAopnM','ACopnm','PAopnm',\
                     'FOopnM','PFopnM','FOopnm','PFopnm']


# %% Tiebar analysis: Overload counting

# Initialize counters for overloads
cnt_overload_5to10 = 0
cnt_overload_10to15 = 0
cnt_overload_15plus = 0
```

```python

# Initialize an empty DataFrame to store rows that meet the conditions
tiebar_overloads = pd.DataFrame()

# Loop through tiebars and filter rows
for column in tiebars:
    condition_5to10 = (machinedata[column] > closing_force / 4 * 1.05) & (machinedata[column] <= closing_force / 4 * 1.1)
    condition_10to15 = (machinedata[column] > closing_force / 4 * 1.1) & (machinedata[column] <= closing_force / 4 * 1.15)
    condition_15plus = machinedata[column] > closing_force / 4 * 1.15

    # Increment counters based on conditions
    cnt_overload_5to10 += condition_5to10.sum()
    cnt_overload_10to15 += condition_10to15.sum()
    cnt_overload_15plus += condition_15plus.sum()

    # Append rows that meet the conditions to the selected_rows DataFrame
    tiebar_overloads = tiebar_overloads.append(machinedata[condition_5to10 | condition_10to15 | condition_15plus])

# Reset the index of the selected_rows DataFrame
tiebar_overloads.reset_index(drop=True, inplace=True)

# Save the selected rows to a new CSV file
tiebar_overloads.to_csv(csv_overloads_tiebars, index=False)

# %% Tiebar analysis: How are the tiebars loaded

# average over the defined interval
avg_F01 = machinedata['F01'].tail(analyzed_cycles).mean()
avg_F02 = machinedata['F02'].tail(analyzed_cycles).mean()
avg_F03 = machinedata['F03'].tail(analyzed_cycles).mean()
avg_F04 = machinedata['F04'].tail(analyzed_cycles).mean()
avg_F0tot = machinedata['F0tot'].tail(analyzed_cycles).mean()

# deviation from average
'''
dev_F01 = (avg_F01 - avg_F0tot/4)/(avg_F0tot/4)
dev_F02 = (avg_F02 - avg_F0tot/4)/(avg_F0tot/4)
dev_F03 = (avg_F03 - avg_F0tot/4)/(avg_F0tot/4)
dev_F04 = (avg_F04 - avg_F0tot/4)/(avg_F0tot/4)
'''
dev_F01 = (avg_F01 - (avg_F02 + avg_F03 + avg_F04)/3)/((avg_F02 + avg_F03 + avg_F04)/3)
dev_F02 = (avg_F02 - (avg_F01 + avg_F03 + avg_F04)/3)/((avg_F01 + avg_F03 + avg_F04)/3)
dev_F03 = (avg_F03 - (avg_F01 + avg_F02 + avg_F04)/3)/((avg_F01 + avg_F02 + avg_F04)/3)
dev_F04 = (avg_F04 - (avg_F01 + avg_F02 + avg_F03)/3)/((avg_F01 + avg_F02 + avg_F03)/3)

# Deviation from other tiebars
machine_calculations = pd.DataFrame()
machine_calculations['Cycle-cnt'] = machinedata['Cycle-cnt']
machine_calculations['Date'] = machinedata['UTC-time'].dt.date

round_decimals = 4
machine_calculations['dev2_F01'] = ((machinedata['F01'] - (machinedata['F02'] + machinedata['F03'] + machinedata['F04']) / 3) \
    / ((machinedata['F02'] + machinedata['F03'] + machinedata['F04']) / 3)).round(round_decimals)
machine_calculations['dev2_F02'] = ((machinedata['F02'] - (machinedata['F01'] + machinedata['F03'] + machinedata['F04']) / 3) \
    / ((machinedata['F01'] + machinedata['F03'] + machinedata['F04']) / 3)).round(round_decimals)
machine_calculations['dev2_F03'] = ((machinedata['F03'] - (machinedata['F02'] + machinedata['F01'] + machinedata['F04']) / 3) \
    / ((machinedata['F02'] + machinedata['F01'] + machinedata['F04']) / 3)).round(round_decimals)
machine_calculations['dev2_F04'] = ((machinedata['F04'] - (machinedata['F01'] + machinedata['F02'] + machinedata['F03']) / 3) \
    / ((machinedata['F01'] + machinedata['F02'] + machinedata['F03']) / 3)).round(round_decimals)

#input for boxplot
# Create bins based on the actual values in 'F01'
bins = np.arange(0, 150, 1)/100*(closing_force/4)

# Create a new DataFrame to store the results
boxplot_df = pd.DataFrame(index=range(len(bins)-1))

# Iterate over columns and calculate counts in each bin
for column in tiebars:
    # Count the values in each bin
    counts = pd.cut(machinedata[column], bins=bins, right=False).value_counts(sort=False).values
    boxplot_df[f'counts_{column}'] = counts
# Sum the counts across all columns to get the 'Total_counts'
boxplot_df['Total_counts'] = boxplot_df.filter(like='counts_').sum(axis=1)
# Add the 'ranges' column
boxplot_df['ranges'] = [f'{start}-{end}' for start, end in zip(bins[:-1], bins[1:])]
# Reorganize the columns
boxplot_df = boxplot_df[['ranges', 'counts_F01', 'counts_F02', 'counts_F03', 'counts_F04', 'Total_counts']]

# %% Scatter plot accelerations

closingunit_maxload = pd.DataFrame()

#columns for cycle count and date
closingunit_maxload['UTC-time'] = pd.concat([machinedata['UTC-time'], machinedata['UTC-time']], ignore_index=True, axis=0)
closingunit_maxload['Cycle-cnt'] = pd.concat([machinedata['Cycle-cnt'], machinedata['Cycle-cnt']], ignore_index=True, axis=0)

#columns for max and min acceleration during closing:
closingunit_maxload['acc_closing'] = pd.concat([machinedata['ACclsM'], machinedata['ACclsm']], ignore_index=True, axis=0)
closingunit_maxload['pos_acc_closing'] = pd.concat([machinedata['PAclsM'], machinedata['PAclsm']], ignore_index=True, axis=0)

```

```python
#columns for max and min acceleration during opening:
closingunit_maxload['acc_opening'] = pd.concat([machinedata['ACopnM'], machinedata['ACopnm']], ignore_index=True, axis=0)
closingunit_maxload['pos_acc_opening'] = pd.concat([machinedata['PAopnM'], machinedata['PAopnm']], ignore_index=True, axis=0)

#columns for max and min force during closing:
closingunit_maxload['F_closing'] = pd.concat([machinedata['FOclsM'], machinedata['FOclsm']], ignore_index=True, axis=0)
closingunit_maxload['pos_F_closing'] = pd.concat([machinedata['PFclsM'], machinedata['PFclsm']], ignore_index=True, axis=0)

#columns for max and min force during opening:
closingunit_maxload['F_opening'] = pd.concat([machinedata['FOopnM'], machinedata['FOopnm']], ignore_index=True, axis=0)
closingunit_maxload['pos_F_opening'] = pd.concat([machinedata['PFopnM'], machinedata['PFopnm']], ignore_index=True, axis=0)

# %% Closing unit analysis: Overload counting

# Initialize counters for overloads
cnt_FrameOverload_Force = 0
cnt_FrameOverload_Acceleration = 0

# Initialize an empty DataFrame to store rows that meet the conditions
frame_overloads = pd.DataFrame()

# Check for Force overloads
for sensor in Force_sensors:
    force_condition = (machinedata[sensor] < -250) | (machinedata[sensor] > 350)
    cnt_FrameOverload_Force += force_condition.sum()

    # Shift the condition by one to get the rows before and after
    previous_rows = force_condition.shift(1, fill_value=False)
    next_rows = force_condition.shift(-1, fill_value=False)

    # Combine the conditions to get the rows before, during, and after the condition
    combined_condition_F = force_condition | previous_rows | next_rows

    # Append rows to 'frame_overloads' DataFrame
    frame_overloads = pd.concat([frame_overloads, machinedata[combined_condition_F]])

# Check for Acceleration overloads
for sensor in Acceleration_sensors:
    acceleration_condition = (machinedata[sensor] < -7500) | (machinedata[sensor] > 7500)
    cnt_FrameOverload_Acceleration += acceleration_condition.sum()

    # Shift the condition by one to get the rows before and after
    previous_rows = acceleration_condition.shift(1, fill_value=False)
    next_rows = acceleration_condition.shift(-1, fill_value=False)

    # Combine the conditions to get the rows before, during, and after the condition
    combined_condition_a = acceleration_condition | previous_rows | next_rows

    # Append rows to 'frame_overloads' DataFrame
    frame_overloads = pd.concat([frame_overloads, machinedata[combined_condition_a]])

# Drop duplicate rows in 'frame_overloads'
frame_overloads = frame_overloads.drop_duplicates()

# Reset the index of the selected_rows DataFrame
frame_overloads.reset_index(drop=True, inplace=True)

# Create new columns in 'frame_overloads' for max force and max acceleration
frame_overloads['max_force'] = frame_overloads[Force_sensors].apply(lambda row: max(abs(row)), axis=1)
frame_overloads['max_acceleration'] = frame_overloads[Acceleration_sensors].apply(lambda row: max(abs(row)), axis=1)

# Select the required columns for 'frame_overloads_table'
frame_overloads_table = frame_overloads[['Cycle-cnt', 'UTC-time', 'max_force', 'max_acceleration']].copy()

# Save the selected rows to a new CSV file
frame_overloads_table.to_csv(csv_overloads_frame, index=False)

# %% Alarm loggings

# Deleting rows with value 5294 in 'AlarmIndex' column
loggings_alarms = loggings_alarms[loggings_alarms['AlarmIndex'] != 5294]

# Delete rows with NaN
loggings_alarms = loggings_alarms[~loggings_alarms['Type'].isna()]

# Reset index
loggings_alarms.reset_index(drop=True, inplace=True)

loggings_alarms.to_excel('loggings_alarms_print.xlsx', index=False)
```

```python
# %% productivity KPIs

production_KPIs = pd.DataFrame()
production_KPIs['date'] = machinedata['UTC-time'].dt.date.unique()

# Count the number of rows per day and add the 'cycles' column
production_KPIs['cycles'] = machinedata.groupby(machinedata['UTC-time'].dt.date).size().values -1

# Add the 'products' column
production_KPIs['products'] = machinedata.groupby(machinedata['UTC-time'].dt.date)['Cycle-cnt'].agg(lambda x: x.max() - x.min()).values

# Initialize an empty list to store valid differences
avg_cycle_times = []

# Loop through each day
for day in production_KPIs['date']:
    # Filter data for the current day
    filtered_data = machinedata[machinedata['UTC-time'].dt.date == day]

    # Calculate differences using vectorized operations
    diffs = filtered_data['Msec'].diff().loc[lambda x: (x >= 5000) & (x <= 60000)]

    # Calculate the mean of valid differences
    avg_cycle_time = round(diffs.mean()) if not diffs.empty else 0

    # Append the result to the list
    avg_cycle_times.append(avg_cycle_time)

# Add the 'avg cycle time' column to production_KPIs
production_KPIs['avg cycle time'] = avg_cycle_times

# Calculate the percentage of a full day
full_day_seconds = 24 * 60 * 60
production_KPIs['uptime'] = round( ((production_KPIs['cycles'] * production_KPIs['avg cycle time']) / 1000) / full_day_seconds * 100)
production_KPIs['uptime'] = production_KPIs['uptime'].clip(upper=100)

production_KPIs['downtime'] = 100 - production_KPIs['uptime']

# Create a new DataFrame alarm_loggings_stops
alarm_loggings_stops = loggings_alarms[loggings_alarms['Status'] == 1][['UTC-time', 'Cycle-cnt', 'AlarmIndex', 'Context']].copy()

# Round 'UTC-time' to date
alarm_loggings_stops['UTC-time'] = alarm_loggings_stops['UTC-time'].dt.date

# Remove duplicates based on all columns
alarm_loggings_stops.drop_duplicates(inplace=True)

# Keep only the first appearance of each unique cycle
alarm_loggings_stops.drop_duplicates(subset=['Cycle-cnt'], inplace=True)

# Change 'Cycletime overrun' to 'Waiting on robot' in 'context' column
alarm_loggings_stops['Context'] = alarm_loggings_stops['Context'].replace('"Cycletime overrun"', '"Waiting on robot"')

# Delete rows with AlarmIndex 2911
alarm_loggings_stops = alarm_loggings_stops[alarm_loggings_stops['AlarmIndex'] != 2911]

# Count the occurrences of each context
context_counts = alarm_loggings_stops['Context'].value_counts()

# Create a new DataFrame with 'context' and 'appearances' columns
alarm_counter = pd.DataFrame({'context': context_counts.index, 'appearances': context_counts.values})

# %% write results in csv file

machine_calculations.to_csv(csv_analysis_calculations, index=False)
closingunit_maxload.to_csv(csv_analysis_accelerations,index=False)
alarm_counter.to_csv(csv_production_alarms, index=False)
production_KPIs.to_csv(csv_production_KPIs, index=False)

# Your existing data
data = [
    ['machine_number', f'X{machine_number}'],
    ['closing_force', closing_force],
    ['last_log_date', last_log_date],
    ['cycle_count', cycle_count],
    ['analyzed_cycles', analyzed_cycles],
    ['cnt_overload_5to10', cnt_overload_5to10],
    ['cnt_overload_10to15', cnt_overload_10to15],
    ['cnt_overload_15plus', cnt_overload_15plus],
    ['avg_F01', round(avg_F01, 0)],
    ['avg_F02', round(avg_F02, 0)],
    ['avg_F03', round(avg_F03, 0)],
    ['avg_F04', round(avg_F04, 0)],
    ['avg_F0tot', round(avg_F0tot, 0)],
    ['dev_F01', round(dev_F01, 3)],
    ['dev_F02', round(dev_F02, 3)],
    ['dev_F03', round(dev_F03, 3)],
    ['dev_F04', round(dev_F04, 3)],
    ['cnt_FrameOverload_Force',cnt_FrameOverload_Force],
    ['cnt_FrameOverload_Acceleration',cnt_FrameOverload_Acceleration],
    ['OEE', 0.74],
    ['Machine Availability', 0.99],
    ['Edge systems availability', 0.89],
    ['Capacity Utalization', 0.85],
    ['Scrap rate', 0.01],
]
```

```
# Convert the list of lists to a Pandas DataFrame
df = pd.DataFrame(data)
df = df.transpose()
df.columns = df.iloc[0]
df = df[1:]
df.to_csv(csv_analysis_results,index=False)

```