TURUN YLIOPISTO

Developing an Industrial Maintenance Management System Using a No-Code Platform

A Step Towards a Value-Adding Internal Service

Mechanical Engineering, Smart Systems
Department of Mechanical and Materials Engineering
Faculty of Technology
University of Turku

Author:
Marco Lafif

05.05.2026
Turku

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

AI-assisted tools such as ChatGPT and Gemini have been used to translate text, check grammar and spelling, as well as to improve the linguistic clarity of the text. The author has edited the text where necessary and takes full responsibility for the content of the work.

Master's thesis

Subject: Mechanical Engineering, Smart Systems
Author: Marco Lafif
Title: Developing an Industrial Maintenance Management System Using a No-Code Platform
Supervisor: PhD (IE), DSc Jussi Kantola
Number of pages: 85 pages
Date: 05.05.2026

As global competition intensifies, particularly for small and medium-sized enterprises (SMEs), optimizing resources and automating manual tasks, such as data collection, has become essential. In the manufacturing industry, production equipment represents critical assets, yet their operation is often insufficiently monitored. Maintenance is a key industrial function with a significant impact on asset management and daily operational performance. By systematically measuring and documenting relevant data, companies can achieve cost savings and optimize operations.

Digitalization offers the potential to improve productivity by enabling automated workflows that execute, document, and report tasks otherwise performed manually. Low-code and no-code platforms enable users to create applications and automated workflows without extensive programming experience, providing SMEs with limited resources an opportunity to modernize processes and enhance operational efficiency.

This thesis investigates which maintenance data should be collected and evaluates the feasibility of designing and implementing a computerized maintenance management system (CMMS) on a no-code platform. The theoretical framework reviews maintenance strategies, introduces axiomatic design as a methodology for software development and system architecture, and provides an overview of no-code platforms. The study examines the client's existing maintenance processes and requirements. An ideal model is developed and serves as the basis for the new CMMS. The resulting application is tested according to a structured test plan to assess the viability and functionality of a no-code-based application for maintenance activities.

The study identified improvement areas in the client's maintenance operations. By applying a systematic approach to address the stakeholder needs, a system architecture was designed, and the CMMS was built on a no-code platform. The test results indicate that the no-code-based CMMS can be used to support maintenance management in SMEs, providing a practical, accessible, and adaptable solution for enhancing workflow efficiency, data collection, automated processing, and decision making.

Keywords: No-Code Platform, Process Development, Productivity, Workflow Automation, AppSheet, Mobile Application, Software Development, GCP, CMMS, Axiomatic Design, Organizational Communication, Industrial Maintenance, Operations Management, TPM, KPI, Internal Service, SME

AI-assisted tools such as ChatGPT and Gemini have been used to translate text, check grammar and spelling, as well as to improve the linguistic clarity of the text. The author has edited the text where necessary and takes full responsibility for the content of the work.

Diplomityö

Koulutusohjelma: Konetekniikka, Älykkäät järjestelmät
Tekijä: Marco Lafif
Otsikko: Teollisen kunnossapidon tietojärjestelmän kehittäminen no-code-alustalla
Ohjaaja: PhD (IE), DSc Jussi Kantola
Sivumäärä: 85 sivua
Päivämäärä: 05.05.2026

Globaalin kilpailun kiristyessä resurssien optimoinnista ja manuaalisten tehtävien, kuten tiedonkeruun, automatisoinnista on tullut välttämätöntä. Tuotantolaitteet edustavat valmistavassa teollisuudessa kriittistä käyttöomaisuutta, vaikka niiden toimintaa seurataan usein puutteellisesti. Kunnossapito on keskeinen teollinen toiminto, jolla on merkittävä vaikutus käyttöomaisuuden hallintaan sekä päivittäiseen operatiiviseen suorituskykyyn. Mittaamalla ja dokumentoimalla keskeistä tietoa järjestelmällisesti yhtiöt voivat saavuttaa kustannussäästöjä sekä optimoida toimintaansa.

Digitalisaatio tarjoaa mahdollisuuden parantaa tuottavuutta automatisoimalla työnkulkuja, jotka suorittavat, dokumentoivat ja raportoivat tehtäviä, jotka muuten toteutettaisiin käsin. Low-code- ja no-code-alustat tekevät sovellusten ja automatisoitujen työnkulkujen luomisen mahdolliseksi ilman laajaa ohjelmointikokemusta. Tämä tarjoaa rajallisilla resursseilla toimiville pk-yrityksille tilaisuuden prosessien modernisointiin ja operatiivisen tehokkuuden parantamiseen.

Tämä diplomityö tutkii, millaista kunnossapidon dataa tulisi kerätä, sekä arvioi kunnossapidon tietojärjestelmän suunnittelun ja käyttöönoton toteutettavuutta no-code-alustalla. Teoreettinen viitekehys käsittelee kunnossapitostrategioita, esittelee aksiomaattisen suunnittelun metodologian sovellus- ja järjestelmäarkkitehtuurin kehittämiseksi sekä tarjoaa katsauksen no-code-alustoihin. Tutkimuksessa analysoidaan asiakkaan olemassa olevia kunnossapitoprosesseja ja vaatimuksia. Näiden pohjalta kehitetään ideaalimalli, joka toimii uuden kunnossapidon tietojärjestelmän perustana. Lopullinen sovellus testataan jäsennellyn testisuunnitelman mukaisesti, jotta no-code-pohjaisen sovelluksen toteutettavuutta ja toiminnallisuutta kunnossapitotarkoituksiin voidaan arvioida.

Tutkimus tunnisti kehityskohteita asiakkaan kunnossapitotoiminnoissa. Järjestelmällisellä lähestymistavalla vastattiin sidosryhmien tarpeisiin suunnittelemalla järjestelmäarkkitehtuuri ja toteuttamalla kunnossapidon tietojärjestelmä no-code-alustalle. Testitulokset osoittavat, että no-code-pohjaista kunnossapidon tietojärjestelmää voidaan käyttää tukemaan kunnossapidon hallintaa pk-yrityksissä. Se tarjoaa käytännöllisen, saavutettavan ja mukautuvan ratkaisun työnkulkujen tehokkuuden, tiedonkeruun, automatisoidun käsittelyn ja päätöksenteon parantamiseksi.

Avainsanat: No-Code alusta, Prosessikehitys, Tuottavuus, Työnkulun automatisointi, AppSheet, Mobiilisovellus, Ohjelmistokehitys, GCP, Kunnossapidon tietojärjestelmä, Aksiomaattinen suunnittelu, Organisaatioviestintä, Teollinen kunnossapito, Operaatioiden johtaminen, Kokonaisvaltainen tuottava kunnossapito, Suorituskyvyn mittaus, Sisäinen palvelu, Pk-yritykset

Tekoälyavusteisia työkaluja, kuten ChatGPT:tä ja Geminiä, on käytetty tekstin kääntämiseen, kieliopin ja oikeinkirjoituksen tarkistamiseen sekä tekstin kielellisen selkeyden parantamiseen. Tekijä on muokannut tekstiä tarvittavilta osin ja kantaa täyden vastuun työn sisällöstä.

Contents

List of Figures

List of Tables

List of Equations

Abbreviations

CA	Customer Attribute
CMMS	Computerized Maintenance Management System
DP	Design Parameter
FR	Functional Requirement
KPI	Key Performance Indicator
LCC	Lifecycle Cost
MP	Maintenance Prevention
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
OEE	Overall Equipment Effectiveness
PM	Preventive Maintenance
PdM	Predictive Maintenance
PV	Process Variable
RCM	Reliability-Centered Maintenance
RFID	Radio Frequency Identification
RM	Reactive Maintenance
TPM	Total Productive Maintenance

1 Introduction

Finland's low productivity growth in recent decades (Calligaris, et al., 2023) and *"the declining trend of well-being at work"* (Hakanen & Kaltiainen, 2024) have been recurring themes in the Finnish national media. Digitalization has been presented as a solution to improve *"European firms' productivity"* (Anderton, et al., 2023). One study shows that service-sector companies investing in software and database assets experienced higher productivity growth, whereas manufacturing firms did not achieve similarly strong outcomes (Koski & Fornaro, 2024).

Despite the modest outcomes of productivity growth measures in the manufacturing sector, this thesis focuses on the development of a database system for a small-sized enterprise operating within the sector. The aim of this thesis is to develop a maintenance management system based on a no-code platform. The system will be designed to meet the client's needs and to improve daily operations by freeing up resources for more value-adding activities, collecting relevant data for future decision-making and enabling the easy implementation of measures that enhance transparency for all stakeholders.

1.1 Background for the Study

Productivity in Finland has not increased as competitively since the 2010s as in the other Nordic countries (OECD, 2024).

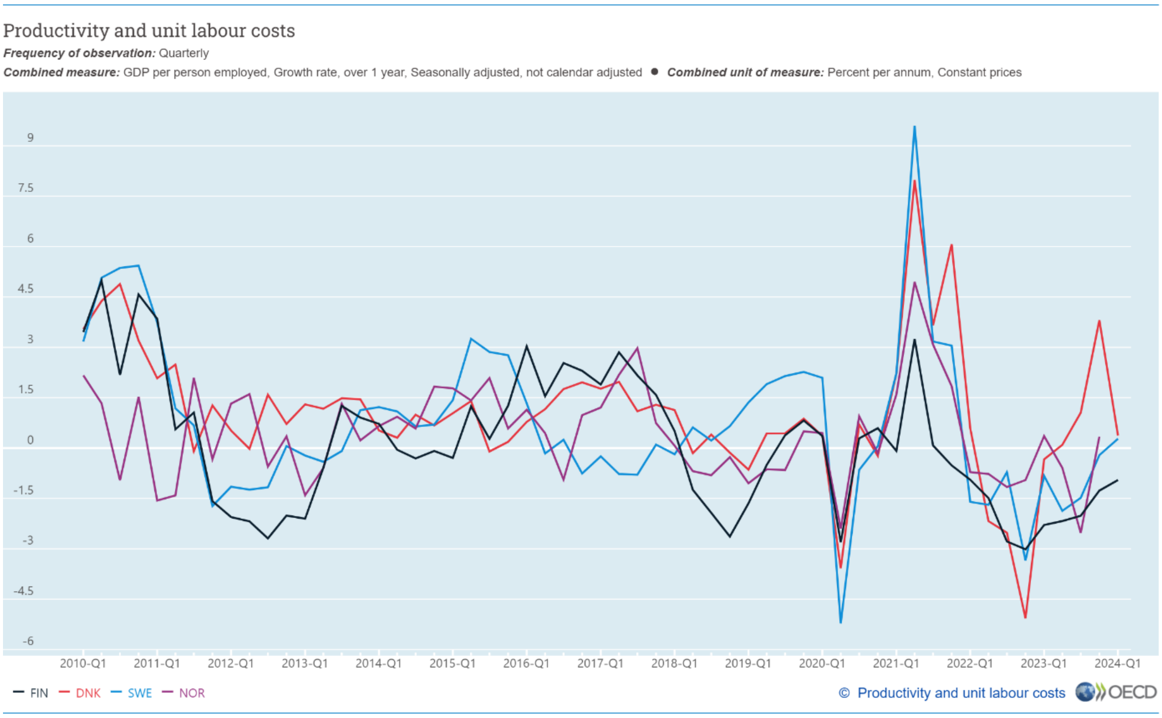


Figure 1 - OECD (2024), "Productivity and unit labour cost by industry, ISIC Rev. 4", OECD Productivity Statistics (database), https://doi.org/10.1787/data-00687-en (accessed on 14 September 2024).

Evolving quality standards, together with a regulated operating environment, require globally operating small-sized enterprises to allocate already scarce resources to data management. Without economically feasible and interdepartmentally integrated data management systems, operators must gather data manually and may also need to duplicate datasets across different databases, increasing overhead and the risk of human error in data creation. To address these challenges, enterprises should invest in new data management or automated data integration systems; however, their cost may be prohibitive, especially for small-sized enterprises. When several non-value-adding daily processes persist (e.g. manually prepared reports that could be automated with existing technology), this can be assumed to have a negative effect on productivity.

Some business collaboration and office tool providers, such as Microsoft and Google, have developed no-code (i.e. low-code) platforms that allow customers to build data management applications without coding expertise (Google, 2024; Microsoft, 2024). These platforms make it possible to integrate data required for daily operations and to automate workflows. For small-sized companies already using these providers' tools, it should be feasible to develop cost-effective data management systems tailored to their needs. This could help enterprises free up resources for more productive tasks.

In the manufacturing industry, *maintenance* is one of the functions – often described as a support function - required to ensure the continuous operation of production equipment and to enhance productivity (Stevenson, 2021, p. 16 & 53). Maintenance can be managed through a maintenance management system, which is part of a "management information system" (Stevenson, 2021, p. 12). Such a system is needed to maintain assets in proper condition throughout their lifecycle and to collect data on production efficiency and costs, enabling a statistical approach to the operational decision making (Järviö & Lehtiö, 2012, p. 185). A maintenance management system not only serves the equipment but can also be developed to support the organization of the maintenance resources (personnel) and the scheduling of maintenance-related tasks. These considerations establish the need to study the feasibility of using a no-code platform to develop a maintenance management system for a small sized enterprise.

1.2 Research Problem

This study is conducted for a small-sized enterprise, hereafter referred to as *the client*, operating in the manufacturing industry. The client has an existing maintenance system that no longer meets their current needs. The system consists of manually filled data sheets containing information about production equipment, pre-scheduled maintenance programs, preset parameters, spare parts and repair history. According to the current process description, the maintenance department is required to carry out planned maintenance tasks in accordance with the programs specified in the equipment information sheets. In addition, the client aims to upgrade its quality management system to comply with the IATF 16949 standard, which requires certified companies to collect performance data on production equipment for use in risk assessment and decision-making (Pačaiová & Ižaríková, 2019, pp. 50 - 51).

The current maintenance process and management systems no longer meet existing requirements. Major line stoppages and failures are recorded manually by the maintenance technicians into equipment information sheets, which creates a risk of human error and misleading information.

There is also a risk on the operating side: when a failure or stoppage occurs, it is typically reported by the operator via a phone call to the maintenance technicians. As a result, no record is created in the system unless the technician manually enters the information. Currently, the only way to document such events in the equipment data sheets is to first record notes in a notebook on site and, after completing the repair, enter the event into the "failure log" tab using a computer in the maintenance office. In cases of minor failures or when higher-priority tasks are ongoing, incidents may go unreported, leading to insufficient information on the operational performance of the production.

1.3 Research Objectives and Scope

Four research questions define the scope of this study:

RQ1. What type of data should the maintenance system collect?

RQ2. How can the development of a maintenance system benefit from the axiomatic design approach?

RQ3. To what extent can a no-code platform be used to develop a CMMS by users without coding knowledge?

RQ4. In what ways does the no-code-based CMMS add value to the daily operations?

The theoretical framework is structured into three subcategories. Maintenance theory literature will be reviewed (Ch. 2.1 – Maintenance), and the client's existing maintenance process descriptions and actions (Ch. 3 – Methods and Datasets) will be investigated. The axiomatic design approach (Ch. 2.2 – Axiomatic Design) will be used to develop an ideal model (Ch. 4 – Maintenance System Development – Ideal Model) for the new maintenance system. In addition to maintenance and axiomatic design theories, the no-code platform will be briefly introduced (Ch. 2.3 – No-Code Platform).

As a result, a maintenance system application will be designed (Ch. 6.1 – Pilot CMMS Design with AppSheet) and tested (Ch. 6.2 – Test Results of the CMMS). The aim is to create a pilot for maintenance management system, which will be implemented alongside a new assembly line. The new system should enable operators to report equipment maintenance needs and failures via mobile devices, increase the amount of definite equipment performance data to support management decision-making, reduce human error in reporting, create a real-time inventory of spare parts, and support maintenance resource management by prioritizing technicians' tasks.

1.4 Research Approach

To assess the feasibility of a new maintenance management system to be built using a no-code platform, this study adopts a mixed-methods approach to investigate and develop the client's maintenance operations. The qualitative phase comprises a literature review, daily operational observations, report analysis, and stakeholder consultations. This phase aims to identify stakeholder needs related to maintenance operations and asset management, as well as to explore the application of the axiomatic design approach in system development. The insights from the empirical phase guide the further development of the maintenance process and the design of the new dedicated system.

Subsequently, a quantitative approach is applied to develop a pilot maintenance management system and evaluate its performance. The system is tested through its application in maintenance data collection. Maintenance performance measures, such as the amount of data collected and time spent, will be benchmarked against the previous measures prior to the adoption of the new maintenance management system.

2 Theory and Literature Review

2.1 Maintenance

To maintain a high level of productivity, a production unit must effectively manage its assets. This requires continuous monitoring, control and development of the manufacturing resources. Without adequate data on production performance, it can be difficult to define the requirements for profitable operations. Therefore, a maintenance management system is needed to collect relevant data to support future decision-making.

Maintenance is often understood as the repair of production equipment. However, maintenance plays a crucial role in keeping assets in good condition and preventing breakdowns. Over the years, multiple strategies and methodologies have been introduced to maintain the production environment and ensure seamless production and equipment availability. These strategies are discussed in this section as maintenance types, which vary in their characteristics. Maintenance types can be divided into reactive and proactive strategies (Järviö & Lehtiö, 2012, pp. 15, 46 - 48).

Depending on available resources, equipment knowledge and company size, enterprises should determine which maintenance types to adopt in order to achieve the most economically sustainable solution over the entire lifecycle of their equipment. Different maintenance types can be combined, and strategies are typically selected based on an enterprise's specific needs and available maintenance history data. Some maintenance strategies require higher initial investment, but they may yield greater long-term benefits by improving production efficiency. (Hamasha, et al., 2023)

Next, some of the globally used maintenance types, such as *reactive maintenance*, *preventive maintenance*, *predictive maintenance*, *reliability-centered maintenance* and *total productive maintenance*, will be introduced. After the introduction of these maintenance types, "*key performance indicators*" (KPI) such as MTBF, MTTR and OEE will be explained.

2.1.1 Reactive Maintenance, RM

"*Reactive maintenance*" (RM) refers to so called *"repair"* or *"run-to-failure (RTF)"* maintenance strategies. Reactive maintenance acts when equipment failure or breakdown has already occurred, reaching a state where the equipment can no longer produce goods as planned. Such maintenance strategy may be used by companies with limited maintenance resources or, in cases where failures are considered to have only a minor impact on production. It may also be chosen when the cost of an unexpected breakdown is estimated to be lower than the cost of proactive maintenance. While the primary benefit is the avoidance of frequent upkeep costs, the downside is an increased risk of unexpected breakdowns occurring at any time. Without proactive measures, these breakdowns can occur in any part of the equipment, leading to unforeseen consequences across the entire production chain. (Alshakhshir & Howell, 2021, pp. 22 - 23; Luthra, et al., 2021, pp. 152 - 153; Tuominen, 2010, p. 87.)

2.1.2 Preventive Maintenance, PM

"*Preventive maintenance*" (PM) aims to reduce the probability of equipment failures and breakdowns. It is a proactive strategy in which maintenance actions are planned in advance based on specific time intervals, usage levels (e.g., number of cycles), or operating hours. In other words, preventive maintenance is performed periodically. The required frequency of preventive actions can be determined based on manufacturer recommendations or by identifying specific characteristics under which maintenance should be performed (e.g., through periodic inspections, lubrication, and functional checks during normal operation.). Typically, the equipment designer or manufacturer defines the recommended maintenance procedures over the equipment's lifecycle. The most economical way to implement preventive maintenance is to identify an optimal point just before a failure is likely to occur, thereby minimizing production losses while keeping the maintenance-related costs – such as personnel and spare parts – as low as possible. It is also important to analyze and compare the costs associated with failures and repairs to those of preventive maintenance actions. (Järviö & Lehtiö, 2012, pp. 50, 53, 96 - 97; Stevenson, 2021, pp. 648 - 649.)

The advantage of preventive maintenance is that resource allocation and task scheduling can be planned in advance, ensuring that all required tools, spare parts and necessary skills are available when needed. Preplanned maintenance actions are considered more efficient than unplanned or reactive actions (Järviö & Lehtiö, 2012, pp. 97, 103 - 106). Furthermore, when a failure occurs, it can be costly due to unpredictable consequences. If actions are not preplanned, this can lead to uncertainties, particularly when the required resources, spare parts, or necessary skills are not readily available on site. (Stevenson, 2021, p. 647).

The disadvantage of preventive maintenance is that it requires resources and personnel for planning and execution, and the allocation of scarce resources can be challenging without proper prioritization. Furthermore, constant periodic inspections, calibrations, part replacements and resource utilization may be considered waste if the timing and frequency of preventive actions are not appropriately defined and adjusted. This can lead to unnecessary costs, particularly if the equipment could continue operating and producing goods without such periodic interventions. (Nguyen, et al., 2008; Stevenson & Kull, 2024, p. 647).

2.1.3 Predictive Maintenance, PdM

"*Predictive maintenance*" (PdM) is a maintenance strategy in which planning and actions are data driven. According to Nguyen, et al. "*A successful predictive maintenance program requires tools and techniques and also expertise of the maintenance personnel to analyze and diagnose the condition of the equipment."* (Nguyen, et al., 2008). The objective is to predict potential equipment failures by monitoring and analyzing real-time data, utilizing reference values, parameters, service history, and past experiences. Modern machines can be equipped with sensors and monitoring systems that detect deviations from normal operating conditions, thereby indicating potential impending failures. By identifying potential failure points based on their frequency of occurrence, preventive measures can be implemented before a breakdown occurs. (Stevenson, 2021, pp. 257, 649).

A common measure of failure frequency, "*mean time between failures*" (More in Ch. 2.1.6 – Mean Time Between Failures, MTBF), is used to assess equipment "*reliability*" and can serve as a tool in predictive maintenance (Stevenson, 2021, pp. 178 - 179). Production equipment and its components should be assessed based on their criticality to operations, which involves evaluating the potential losses caused by a failure. According to Bartosz (Lean Community: M. Bartosz, 2021) there is a common trend to use 85% of the MTBF as a threshold for replacing "key parts" to avoid failures with major impacts.

The advantage of predictive maintenance compared to preventive maintenance is that it can generate cost savings by enabling component replacement just before a breakdown occurs, thereby allowing maximum utilization of parts. In contrast, preventive maintenance requires periodic replacement even when components are still in good operational condition. The disadvantage of predictive maintenance is that it requires sufficient data to make reliable predictions about equipment performance. This may be difficult to achieve for organizations with limited resources. (Stevenson, 2021, p. 649).

2.1.4 Reliability-Centered Maintenance, RCM

"*Reliability-centered maintenance*" (RCM) is a maintenance method that originated in the airline industry. Its developers investigated failure modes in aircraft, as it was previously assumed that operational reliability could be improved through periodic overhauls. This preventive maintenance approach was costly for airlines, and it was observed that some failures could not be prevented. It was also shown that the probability of failure was not necessarily related to elapsed time, and different failure patterns were identified. (Järviö & Lehtiö, 2012, p. 79; 162; Nowlan & Heap, 1978, pp. 2 - 4.)

The RCM method has been further developed to make it more suitable for the manufacturing industry. First, the level of implementation of the RCM method must be determined, meaning which assets the method will be applied to. Processes and equipment are prioritized based on cost, safety, environmental and quality perspectives. The primary objective of RCM is to define the functions of the equipment, specifying what it must achieve within its operating environment. This guides both maintenance and operator objectives. Once the functions are defined, all potential failure states, their conditions, and related events must be identified. The potential causes of failures are then used as input for "f*ailure mode and effects analysis*" (FMEA*)*. In FMEA, the probability and consequences of failures are determined. Together, probability and consequence form the risk. The aim is to reduce this risk. This can be achieved through condition monitoring, periodic repair or replacement, failure-finding tasks, or equipment redesign. (Basson & Moubray, 2018; Järviö & Lehtiö, 2012, pp. 163 - 167.)

The advantage of RCM is the potential to increase equipment reliability when a sufficient risk assessment of processes and equipment is conducted. It provides a systematic approach to supporting decision-making (e.g., evaluating the requirements for necessary spare parts). RCM also helps direct resources to where they are most needed. The disadvantage is that RCM requires resources to be implemented as intended. (Basson & Moubray, 2018, pp. 473 - 474; Järviö & Lehtiö, 2012, p. 159).

2.1.5 Total Productive Maintenance, TPM

"*Total productive maintenance*" (TPM) is a maintenance strategy that aims to maximize "*overall equipment efficiency*" (OEE). TPM was developed by Seiichi Nakajima in the 1970s in Japan (Nakajima, 1988). TPM aims to develop a system and culture in which every stakeholder is committed to contributing to the efficient operation of equipment and processes (Järviö & Lehtiö, 2012, p. 146).

There is no single way to implement TPM as a maintenance strategy, as companies' needs and operating environments vary. However, TPM can be divided into five key developmental activities (Nakajima, 1988, pp. 49 - 52):

1. "Eliminating six big losses"
2. "Autonomous maintenance program"
3. "Scheduled maintenance program"
4. "Training of operations and maintenance personnel"
5. "Equipment management program"

The first TPM developmental activity is to recognize waste and reduce the six big losses, which are "*equipment failures*", "*setup and adjustment time*", "*idling and minor stoppages*", "*reduced speed*", "*process defects*", and "*reduced yield*" (Nakajima, 1988, pp. 14, 50).

The second activity is to establish an autonomous maintenance program. Operators should be encouraged to actively participate in maintenance, as they work with the equipment continuously and can more easily detect deviations (e.g. abnormal sounds or missing visual features compared to normal operation). TPM assigns users *"-- to be responsible of their own equipment"* (Nakajima, 1988, pp. 72 - 73). Operators can be trained to perform daily inspections and to keep the equipment clean, as dirt can cause component wear. Over time, operators' knowledge of the equipment increases, and they can be trained to perform more challenging maintenance tasks. This also improves their understanding of equipment functionality and enables them to take broader responsibility for its upkeep, thereby freeing maintenance personnel for more demanding tasks. It is important to monitor the performance of autonomous maintenance and report the results so that the activity is also perceived as meaningful from the operators' perspective. (Järviö & Lehtiö, 2012, pp. 152 - 158.)

Creating a scheduled maintenance program for the maintenance department is the third key activity of TPM. The scheduled maintenance program is based on the previously mentioned planned maintenance strategies for equipment (Ch. 2.1.2 – Preventive Maintenance and 2.1.3 – Predictive Maintenance), (Nakajima, 1988, p. 55).

Equipment users can cause accidental errors if they are not properly trained (Nakajima, 1988, p. 38). To prevent user errors, companies implementing TPM should train their employees to better understand and operate their equipment. In addition to training in equipment operation, operators should also be trained in maintenance techniques (e.g., starting from daily inspections). Furthermore, maintenance personnel's skills should be developed through advanced technical training (Nakajima, 1988, p. 94). Various training and educational institutions in Finland offer further training courses and provide certifications for qualified maintenance professionals. Improving the operational and maintenance skills of employees constitutes the fourth development activity of TPM (Nakajima, 1988, p. 90).

The fifth and final development activity of TPM is to establish an early equipment management program. By implementing this activity, it is possible to increase the reliability, maintainability, and safety of equipment and thereby reduce potential costs over the equipment's lifecycle. "*The lifecycle costs*" (LCC) can be influenced at various stages of the equipment's lifecycle (Nakajima, 1988, pp. 96-98):

1. "The equipment investment planning"
2. "Design"
3. "Fabrication"
4. "Installation"
5. "Test Running"
6. "Commissioning"

One of the most important stages mentioned above is design. It is essential to share knowledge and data between the design, maintenance, and operations departments through "*horizontal communication*" in order to achieve highly reliable and easily maintainable equipment. "*Maintenance prevention"* (MP) and "*maintenance-free design"* are enhanced through early equipment management. The design stage sets the boundaries for the equipment's operational, maintenance, and energy costs. During this phase, historical maintenance data can be utilized to eliminate the waste associated with fixing faults after the equipment is commissioned. The key to the early equipment management program is to prevent potential problems during start-up by sharing knowledge among stakeholders, thereby reducing waste caused by detected and subsequently corrected faults. (Nakajima, 1988, pp. 96-101.)

2.1.6 Mean Time Between Failures, MTBF

When equipment failures are documented with their times of occurrence, failure intervals can be identified. By using historical failure data, the average uptime of a system or piece of equipment can be calculated to predict potential future failures. *Mean time between failures* (MTBF) is a metric defined as the total uptime before failures t_u, divided by the number of failures *n* (Tongdan, 2019, pp. 10 - 12):

$$MTBF = \frac{1}{n}\sum_{i=1}^{n} t_{ui} \tag{1}$$

Equation 1 – MTBF.

2.1.7 Mean Time to Repair, MTTR

Mean time to repair (MTTR) measures the average time required to repair specific equipment. MTTR can be defined in a similar way as the MTBF above, by calculating the sum of repair times t_r, divided by the number of repairs *n*:

$$MTTR = \frac{1}{n}\sum_{i=1}^{n} t_{ri} \tag{2}$$

Equation 2 – MTTR.

When repair time is documented, it is important to consider the entire process chain from reporting a failure to restoring the equipment to operational condition. Time is required for technicians to reach the site and diagnose the issue. Spare parts may have lead times, even if they are available in stock. Replacement or repair of components also takes time, and testing, commissioning, and administrative tasks may further delay the return of the equipment to production (Tongdan, 2019).

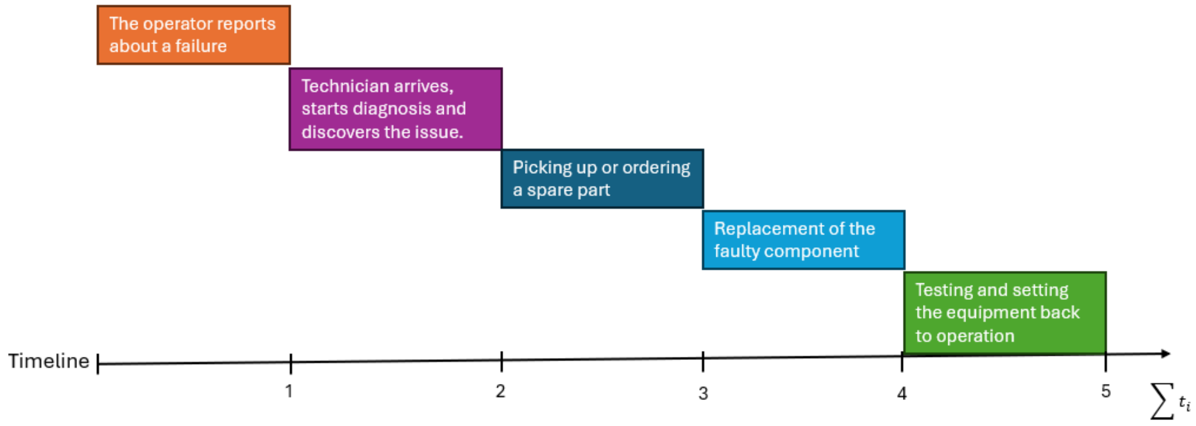


Figure 2 – In an ideal maintenance management system, the total duration of the repair process is documented using MTTR metrics to account for the entire process chain and associated losses. This figure has been derived from an MTTR estimation example and adapted to meet the client's specific operational conditions (Tongdan, 2019, pp. 14, Figure 1.10)

2.1.8 Overall Equipment Effectiveness, OEE

As described earlier in Chapter 2.1.5 (Total Productive Maintenance, TPM), the TPM maintenance strategy aims to maximize OEE. The six big losses are used to determine OEE. The first two losses, equipment failures and setup time, define availability (Nakajima, 1988, pp. 25, Figure 5). "*Availability*" is determined by the rate of failures. By measuring the previously introduced MTBF and MTTR, availability can be expressed as follows (Stevenson & Kull, 2024, pp. 185 - 186):

$$A = \frac{MTBF}{MTBF + MTTR}\ ;\ A = availability \tag{3}$$

Equation 3 – Availability.

In addition to losses in production time, losses in production speed and quality must also be considered. The third and fourth factors among the six big losses are "*idling*" and "*reduced speed*". These are defined by calculating "*performance efficiency*" (P), which is the product of "*operating speed rate*" and the "*net operating rate*". The operating speed rate measures the difference between the designed equipment capacity and actual output (Nakajima, 1988, pp. 24 - 27):

$$Operating\ Speed\ Rate = \frac{theoretical\ cycle\ time}{actual\ cycle\ time} \tag{4}$$

Equation 4 - Operating Speed Rate.

The net operating rate accounts "*minor stoppages*" (Nakajima, 1988, p. 26):

$$Net\ Operating\ Rate = \frac{proccessed\ amount * \ actual\ cycle\ time}{operation\ time} \quad (5)$$

Equation 5 – Net Operating Rate.

Performance efficiency can be derived as the product of equations 4 and 5 (Nakajima, 1988, p. 27):

$$Performance\ Efficiency\ (P) = Operating\ Speed\ Rate * Net\ Operating\ Rate$$

$$= \frac{theoretical\ cycle\ time}{\cancel{actual\ cycle\ time}} * \frac{proccessed\ amount * \cancel{actual\ cycle\ time}}{operation\ time}$$

$$= \frac{theoretical\ cycle\ time * \ proccessed\ amount}{operation\ time} \quad (6)$$

Equation 6 – Performance Efficiency.

"*Defects in process*" and "*reduced yield*" constitute the "*quality rate"* of the OEE. The quality rate is defined as the ratio of conforming products to the total processed amount (Nakajima, 1988):

$$Quality\ Rate\ (Q) = \frac{proccessed\ amount - \ defect\ amount}{processed\ amount} \quad (7)$$

Equation 7 – Quality Rate.

Finally, OEE is defined by integrating the data from the equations 3, 6 and 7, representing the losses in time, speed and quality (Nakajima, 1988, pp. 27 - 28):

$$Overall\ Equipment\ Efficiency\ (OEE) = A * P * Q \quad (8)$$

Equation 8 – OEE.

	Reactive Maintenance - RM	Preventive Maintenance - PM	Predictive Maintenance - PdM	RCM	TPM
Approach	Reactive	Proactive	Proactive	Proactive	Proactive
Cost	While no service costs are budgeted, unexpected costs may arise during breakdowns.	Periodic service and replacement costs, as well as costs related to planning and tools.	Costs related to planning, monitoring, tools and techniques.	Costs associated with continuous risk assessment based on available data, as well as planning, monitoring, tools and	Costs associated with monitoring, reporting, tools, and techniques.
Strategy	Resources are allocated only when repairs are necessary.	Preventing breakdowns through scheduled maintenance and planned component replacement.	Preventing breakdowns through monitoring and data-driven maintenance and replacement.	Risk-based focus on prioritized assets. Preventing breakdowns through monitoring and data-driven maintenance and replacement.	Preventing breakdowns through personnel involvement and commitment, and through monitoring and data-driven maintenance and replacement.
Maintenance interval	-	Scheduled individually for each piece of equipment.	Data-driven approach for each piece of equipment.	Data-driven approach for each piece of prioritized equipment.	Data-driven approach (MTBF, MTTR, OEE).
Resources	-	Technicians, dedicated time and necessary tools.	Resources for equipment monitoring, analytical and diagnostic expertise, and relevant data, tools, and techniques.	Resources for equipment monitoring, expertise on analytics and diagnostics, and relevant data, tools, and techniques.	Operators, technicians, dedicated time, and tools.
Risk	Unexpected downtime and production losses.	If maintenance intervals are set to too long, failures may occur. Conversely, if the intervals are significantly shorter than the optimal period, the costs may outweigh the benefits.	Insufficient data may mislead maintenance activities.	Insufficient data may mislead maintenance activities.	Insufficient data may mislead the maintenance activities. Responsibility for equipment maintenance is distributed across the organization, requiring constant communication.

Figure 3 – Comparison of the introduced maintenance types.

2.1.9 Maintenance Economics

Financial management is a key element in managing business and maintenance activities. Operations must be cost-effective; therefore, revenues must exceed costs. Costs can be divided into "*direct and indirect costs*". Direct costs (e.g. wages, spare parts and other material costs) are more commonly measured than indirect costs (e.g. scrap, rework, inventory value, inefficient use of resources, overtime, and increased lifecycle costs), even though the indirect costs may constitute a larger share of total operational costs. (Järviö & Lehtiö, 2012, pp. 179 - 181.)

One key task of maintenance is to collect information for future decision-making. Historical cost data and maintenance performance statistics (e.g., failures, downtime, scrap, and repair costs) are essential for data-driven management decisions, such as equipment replacement and investment planning. (Järviö & Lehtiö, 2012, pp. 118 - 119; Stevenson & Kull, 2024, p. 660.)

2.2 Axiomatic Design

Since there is a need to design a new maintenance system, *Axiomatic Design* (AD) has been chosen as the method for system development. To ensure a scientific approach to system design, AD provides a systematic way of defining and achieving functions that meet customer needs across various systems. In this thesis, AD is used to develop the "*system architecture*" for a new *computerized maintenance management system* (CMMS). In general, AD aims to reduce design complexity and improve robustness. It also streamlines the design process, reducing the need for multiple prototyping, testing, and iteration phases, thereby saving time and cost. (Suh, 2001, pp. 5 - 6, 57 - 58 & 192 - 195.)

2.2.1 Axiomatic Design Theory

When design work is carried out on a "*trial-and-error*" basis, designers often rely on implicit design goals (Suh, 2001, pp. 2-3). AD provides a way to use "*– – codified and generalized knowledge – –*" to make the design more explicit "*– – by allowing correct decisions to be made quickly and the first time around."* (Suh, 2001, pp. 4 - 5). The goal of AD is to provide designer with a basis "*– – based on logical and rational thought processes and tools*" (Suh, 2001, p. 5). Design can be defined as a combination of statements "*what we want to achieve"* and "*how we will achieve it"* (Suh, 2001, p. 3). AD forms a framework for linking these two statements. It is based on two axioms, *the independence axiom and the information axiom*, which guide the design process.

In general, design characteristics are similar across diverse fields. Designers should begin the problem-solving process by understanding customer needs and defining the problem. A solution concept is then developed through synthesis and optimized through analysis, after which the results should be evaluated to ensure they meet the initial needs. (Suh, 2001, p. 3.) AD provides a theoretical framework for defining these design characteristics, and it can be divided into four domains (Suh, 2001, p. 10):

1. "Customer domain"
2. "Functional domain"
3. "Physical domain"
4. "Process domain"

The characteristics of the domains are defined through the "*mapping process*". First, "*customer needs*" (i.e. "*customer attributes*" [CAs]) are identified in the customer domain. Stakeholders should be properly identified to ensure the correct target group for both the design and the problem definition. Next, in the functional domain, "*functional requirements*" (FRs), which represent the functional needs, and "*constraints*" (Cs), which act as thresholds or boundaries for the design or system, are defined based on the CAs. The creation of FRs should take place in a "*solution-neutral environment"*, meaning that existing designs or solutions should not be considered when defining FRs. In the physical domain, "*design* *parameters*" (DPs) are identified as elements of the design that satisfy the specified FRs. The DPs should comply with the defined constraints. According to the information axiom, the preferred design is the one with minimal information content, as this increases the likelihood of fulfilling the FRs. The last domain is the process domain, where "*process variables*" (PVs) are selected to generate the required DPs. (Suh, 2001, pp. 10 - 14, 58 - 59, 196 - 197.)

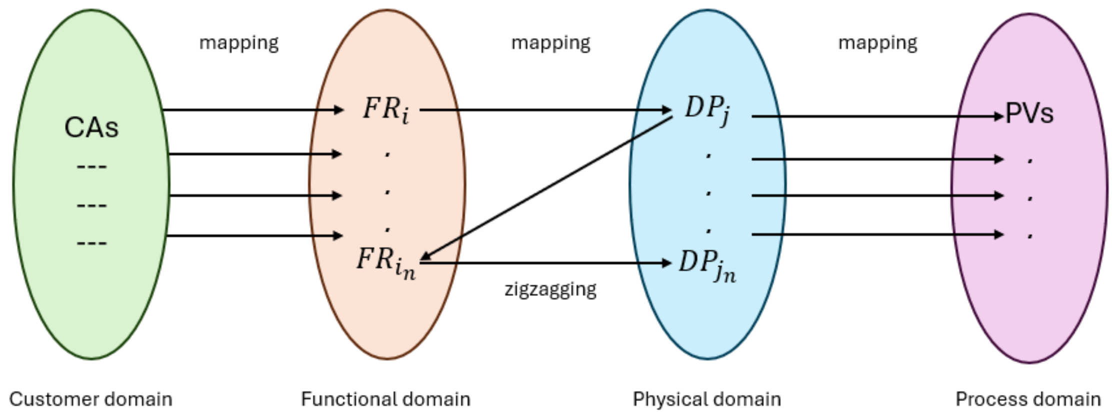


Figure 4 - The four domains of axiomatic design (Suh, 2001, p. 11) are formalized to represent the mapping and zigzagging process.

The FRs, DPs and PVs are decomposed from higher hierarchical levels, representing conceptual information, to more detailed lower levels by "*zigzagging*" between the domains until the design becomes applicable. The outcome of the zigzagging process depends on the designer's experience and knowledge base. (Suh, 2001, pp. 200 - 205.)

2.2.2 Design Matrix

Design equations and the "*design matrix*" [A] can be defined through the mapping process described above. The relationship between the vector of FRs and vector of DPs can be described as follows (Suh, 2001, p. 18):

$$\{FR\} = [A]\{DP\} \tag{9}$$

Equation 9 - Relationship between FRs and DPs.

An example of a (3x3) design matrix with three FRs and three DPs is shown below (Suh, 2001, p. 18):

$$[A] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{10}$$

Equation 10 – 3x3 Design Matrix.

where (Suh, 2001, p. 19):

$$FR_1 = A_{11}DP_1 + A_{12}DP_2 + A_{13}DP_3$$
$$FR_2 = A_{21}DP_1 + A_{22}DP_2 + A_{23}DP_3 \quad (11)$$
$$FR_3 = A_{31}DP_1 + A_{32}DP_2 + A_{33}DP_3$$

Equation 11 - Eq.10 in terms of its elements

The design matrix can be represented with rows corresponding to each FR and columns corresponding each DP, as shown in the following table (Suh, 2001, pp. 225, 273):

	DP1	DP2	DP3
FR1	A11	A12	A13
FR2	A21	A22	A23
FR3	A31	A32	A33

Table 1 - Design matrix is shown as a table representing the relationship between each FR and DP as its elements. Typically, a relationship is indicated in a cell with value “x”, whereas the absence of a relationship – meaning that the partial derivative is zero – is represented by a blank cell or a value of 0. (Suh, 2001, pp. 32 - 33, 282 - 286, 463)

The partial derivative expresses the relationship between each FR and DP. When a change in a DP affects an FR, the partial derivative is not equal to zero (Brown, 2006; Suh, 2001):

$$A_{ij} = \frac{\partial FR_i}{\partial DP_j} \quad (12)$$

Equation 12 – Partial Derivative.

The elements in the design matrix can be described either as functions of DPs (“*nonlinear design*”) or as constants (“*linear design*”) (Suh, 2001, pp. 19 - 20). The aim of the design matrix is to satisfy the “*Independence Axiom*”, which describes the independence of the design functions. This means that, in the case of multiple FRs, the design should be derived so that each FR can be achieved without affecting the others (Suh, 2001, pp. 16 - 17). Correspondingly, DPs should operate independently in relation to each FR. This is why zigzagging is needed when mapping DPs to fulfil the FRs: the process moves from top level to lower levels and back again within the design hierarchy (Suh, 2001, pp. 17, 59). It may be beneficial to have an equal number of FRs and DPs in order to satisfy the independence axiom (Suh, 2001, p. 116). This type of situation results in a square design matrix. While the amount of FRs depends on the system, the equation presented earlier (Eq. 9 on p.17) can be expressed as follows (n = number of DPs):

$$FR_i = \sum_{j=1}^{n} A_{ji}\, DP_j \tag{13}$$

Equation 13 – Eq. 9 in terms of its elements, where “n = number of DPs” (Suh, 2001, p. 19).

To achieve the independence axiom, either a “*diagonal*” or a “*triangular design matrix*” is required. A diagonal design matrix means that the DPs satisfy the FRs independently. This type of design is called an “*uncoupled design*”, meaning that the DPs can be adjusted in any order without affecting the FRs; therefore, it represents an “*ideal design*”. An example of a design matrix with two independent FRs and two DPs is shown below (Suh, 2001, p. 19):

$$[A] = \begin{bmatrix} \boldsymbol{A_{11}} & 0 \\ 0 & \boldsymbol{A_{22}} \end{bmatrix} \tag{14}$$

Equation 14 - Nonzero values form a diagonal in the design matrix. This is called an uncoupled design.

A triangular design matrix is called a “*decoupled design*”. This means that more than one DP affects a single FR. This type of design has a disadvantage: when a DP is changed, it affects the related FRs, thereby introducing constraints into the design. However, it is possible to satisfy the independence axiom by reordering the FRs and DPs into an appropriate sequence. The aim is to arrange the elements so that the zeros are located either above the diagonal, forming “*lower triangular*” (LT) matrix, or below the diagonal, forming an “*upper triangular*” (UT) matrix. (Suh, 2001, p. 19.)

An example of a 3x3 decoupled design matrix, reordered by its FRs and DPs, is shown below:

	DP1	DP2	DP3
FR1	x	0	x
FR2	0	x	0
FR3	0	x	x

Table 2 - A decoupled design has nonzero values on both sides of the diagonal.

	DP2 ←	DP3 ←	DP1 →→
FR2 ↑	x	0	0
FR3 ↑	x	x	0
FR1 ↓↓	0	x	x

Table 3 - The design shown in Table 2 is reordered to form a lower diagonal (LT) matrix.

A coupled design represents a case where changing one DP affects all the other FRs. An example of a coupled design is shown in the table below:

	DP1	DP2	DP3
FR1	x	X	x
FR2	x	X	x
FR3	x	X	x

Table 4 - Coupled design.

The design matrix, as a method, is beneficial for identifying acceptable designs through functional analysis in the early stages of a design (Suh, 2001, p. 59).

2.2.3 System Hierarchy

The AD design philosophy is based on the decomposition of higher-level FRs, DPs, and PVs into lower-level FRs, DPs, and PVs (i.e., from conceptual to detailed design), thereby establishing a system hierarchy. This hierarchy may be represented through "*module-junction diagrams*", "*flow diagrams*", or layered design matrices to construct a "*hierarchical tree*" of decomposed domains (Suh, 2001, pp. 332 - 333). These representation methods facilitate the determination of a sequential order for the design or its operation, representing the overall system architecture. (Suh, 2001, pp. 208 - 213.)

The FRs, DPs, and PVs are decomposed into lower levels until the design can be executed. The nodes at the lowest level are referred to as "*leaves*" (Suh, 2001, p. 209). An example of a system decomposed into three sublayers is presented below:

$$
\begin{aligned}
\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} &= \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \end{Bmatrix} \\
\begin{Bmatrix} FR_{11} \\ FR_{12} \end{Bmatrix} &= \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \begin{Bmatrix} DP_{11} \\ DP_{12} \end{Bmatrix} \\
\begin{Bmatrix} FR_{21} \\ FR_{22} \end{Bmatrix} &= \begin{bmatrix} X & 0 \\ X & X \end{bmatrix} \begin{Bmatrix} DP_{21} \\ DP_{22} \end{Bmatrix} \\
\begin{Bmatrix} FR_{221} \\ FR_{222} \end{Bmatrix} &= \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \begin{Bmatrix} DP_{221} \\ DP_{222} \end{Bmatrix}
\end{aligned}
\tag{15}
$$

Equation 15 - Three-layer decomposition of an example system.

The hierarchy of the example system can be represented as a hierarchical tree illustrating the decomposition (Suh, 2001, pp. 209, 252, 264, 272):

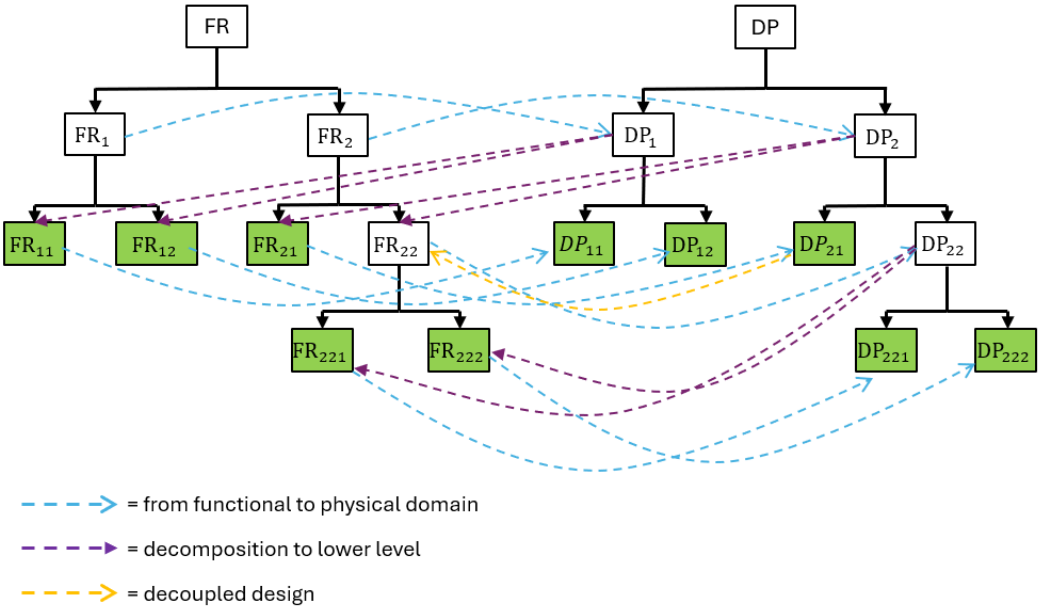


Figure 5 - The hierarchical tree of the example system (Eq. 15) shows the decomposition of FRs and DPs. The leaf nodes of the FRs and DPs are colored green.

Other ways to represent the hierarchy are the module-junction diagram and the flow diagram. In these diagrams, an FR together with its corresponding DP form up a "*module*" (Suh, 2001, p. 209). The modules are combined into a hierarchical structure using circled symbols: a circled letter S (i.e., sum) indicates an uncoupled design, a circled letter C (i.e., control) indicates a decoupled design, and a circled letter F (i.e., feedback) indicates a coupled design. In the junction-module diagram, the sequence starts from the bottom-level leaves and proceeds to higher levels by combining the lower-level modules to obtain higher-level modules, which in turn yield higher-level FRs. (Suh, 2001, pp. 209 - 211.)

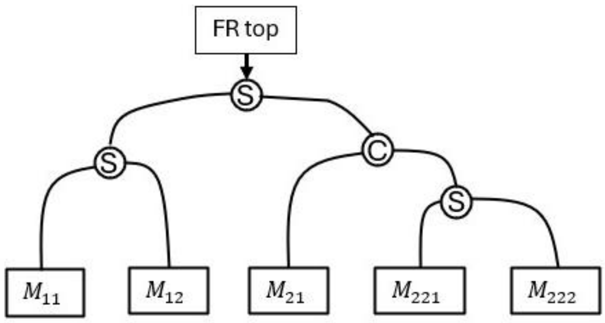


Figure 6 - The module-junction diagram of the example (Eq. 15) shows the system decomposition (Suh, 2001, p. 211).

In a flow diagram, the modules are arranged in a given sequence according to the design matrix. In an uncoupled design, the outputs can be combined in any sequence, whereas in a decoupled design, the left-side module is controlled first and the right-side module afterward. A coupled design forms a feedback junction, requiring the output of the right-side module to be fed back to the left-side module. (Suh, 2001, p. 212.)

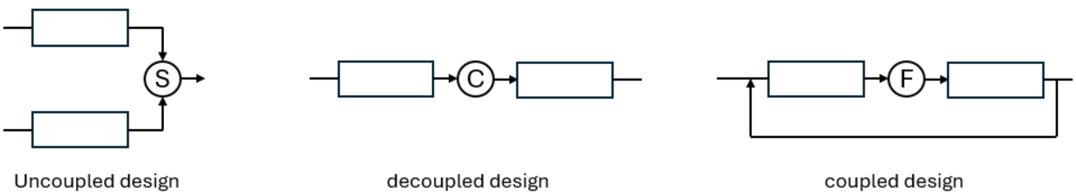


Figure 7 - The boxes in the diagrams above represent the modules. The three different junctions for each design type are shown above (Suh, 2001, p. 211).

The benefit of a flow diagram representation of system architecture is that it structures the system according to its modules and operational sequence. The sequence progresses from left to right (when in series), and the innermost boxes (leaves) are used as inputs for the outermost modules (top level). Thus, the flow diagram operates from the inside outward, ultimately producing the desired output on the right. (Suh, 2001, pp. 211 - 214.)

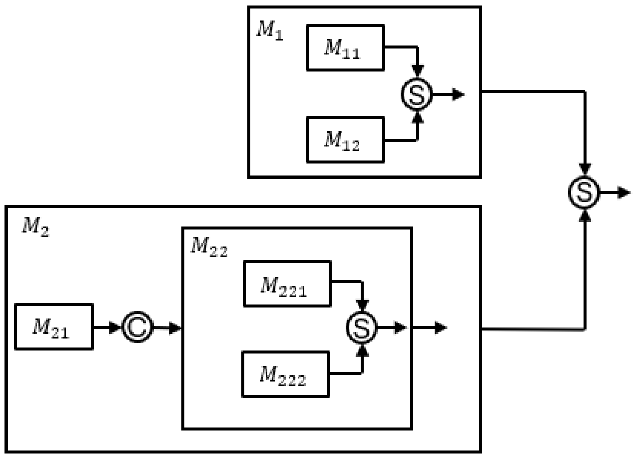


Figure 8 - The flow diagram of the example system (Eq. 15) (Suh, 2001, pp. 212 - 213).

2.3 No-Code Platform

2.3.1 From Low-Code to No-Code Applications

The term "*Low-code applications*" was introduced by Forrester Research Ltd. in 2014 (Forrester Research: Richardson, et al., 2014). Low-code applications are platforms that can be used to develop applications with little or no coding knowledge. This approach to application development has enabled enterprises to establish systems independently and quickly to automate their daily business operations without any dependency on software coding suppliers. Low-code system development provides benefits such as improved "*privacy*", since business information does not need to be shared outside the organization, as well as "*rapidness*" and "*cost reduction*", as end users on the customer side can act as developers themselves, often referred to as "*citizens*" (Sanchis, et al., 2019). The rapid development speed is largely due to the use of the pre-built components and reusable blocks provided by low-code ecosystems, making the approach modular (Viljoen, et al., 2026).

A study by Bodicherla (2025) reports that the scalability of applications can cause difficulties if “*governance frameworks*, *security protocols, and best practices*” are not considered during implementation (Bodicherla, 2025).

Over the years, major business application and cloud service providers such as Amazon, Microsoft and Google have further developed the low-code platforms by introducing application builders such as AWS App Studio, Microsoft Power Apps and Google AppSheet. These platforms enable the generation of applications through visual editors that link and integrate data stored in the cloud. In addition, the providers’ AI services can support application creation by offering “*smart suggestions*” for faster development. (Amazon Web Services, Inc., 2025; Google, 2025; Microsoft, 2025)

The client operates within the Google Workspace environment, so this study focuses on the functionalities and possibilities of Google AppSheet. AppSheet was founded by Praveen Seshadri and Brian Sabino in 2014 to help businesses improve productivity through a mobile application platform (Geekwire: M.Nickelsburg, 2015; GeekWire: S.Taylor, 2020). Its main purpose was to leverage existing spreadsheets as a database that could be shared through a mobile environment. By using spreadsheets as a database, “*non-developers*” (Geekwire: M.Nickelsburg, 2015) can create mobile applications for internal business purposes without coding knowledge. AppSheet was later acquired by Google to support Google Cloud’s strategy of modernizing customers’ application development for business processes through a software platform. (Forbes: A. Bridgwater, 2020.)

2.3.2 Functionality of a No-Code Platform

According to Google, Google AppSheet can be used as a “no-code platform” to automate workflows and to create applications for data collection and integration. The difference between low-code and no-code platforms is that the latter are designed to be used without coding skills. User management and governance can be adapted from existing libraries or configured for applications according to an organization’s own policies. (Google, 2025.)

As mentioned earlier, spreadsheets are used as a database in AppSheet. Existing tables (e.g., Excel or Google sheets) can be used as the basis for application development. AppSheet functions, similar to Excel formulas, are used to control the behavior of row and column information within tables (Google, 2026).

AppSheet also offers the possibility of creating databases from scratch. Depending on the license type, users can create multiple applications, databases, and tables within Google Cloud. The free license allows up to ten test users, 1000 rows / database, five databases, and it provides features for prototyping, testing, and personal app creation (Google, 2026). In addition, there are three different license tiers that offer more features through monthly subscriptions. Some Google Workspace business plans include AppSheet licensing. (Google, 2026)

There are two editor modes available in the AppSheet app editor. The "*legacy editor*" consists of nine different pages that can be accessed from the left sidebar. The "*Home page*" suggests next steps and introduces the "*Learning Center*" with tutorials. The "*Info page*" provides access to the app's properties. The "*Data page*" allows access to the data used within the app. The "*UX page*" contains the settings for the app's views. The "*Behavior page*" allows the creation of actions for the app, while the "*Automation page*" enables the creation of bots. The "*Security page*" controls user access to the app. The "*Intelligence page*" supports the creation of machine learning models for the app. The "*Manage page*" provides version control for the app. The "Improved editor" mode presents the same pages as the legacy editor, but with different naming and layouts. (Google, 2026)

3 Methods and Datasets

Since the aim of this thesis is to develop a new maintenance system, it is first necessary to establish a baseline of the client’s current systems, procedures, and needs. The existing system will be investigated by observing actual operations at the production site, examining process descriptions within the current quality management system, and conducting stakeholder consultations regarding current requirements.

3.1 Observations and Datasets

The study begins with site observations and a review of process descriptions. These observations focus on three key maintenance-related activities: how maintenance needs are communicated and to whom, how maintenance tasks are performed, and how and where documentation is stored. In parallel, stakeholders will be interviewed to determine how the tasks described in the process descriptions are actually carried out. Following this diagnostic phase, the findings will be cross-referenced with the process descriptions to form a comprehensive analysis of the existing system, which will then serve as the foundation for the development of the new CMMS.

3.1.1 Preventive Maintenance Process

According to the maintenance process description, the daily PM tasks, referred to as “daily inspections”, are conducted by operators. These inspections are manually recorded on paper-based documents. The same information must later be scanned, transferred to the client’s database, and copied into the equipment information sheet.

Each piece of equipment has its own PM schedule, which can be found in the equipment information sheet.

Weekly, monthly, quarterly, semi-annual, and annual PM tasks can be tracked using the “monthly calendar of PM tasks” sheet, which indicates whether maintenance-performed PM tasks have been completed for each piece of equipment.

3.1.2 Reactive Maintenance Communication Process

The client’s process description for maintenance outlines the aim, scope, roles, responsibilities, and actions. In the event of a failure, production personnel are required to inform their supervisor of any malfunctions or faults in the equipment. According to the process description, reporting and documentation are the responsibility of the production supervisor.

However, based on the author's observations, the actual procedure typically involves the maintenance technician being informed directly by the operator, while the production supervisor may also be notified in some cases. In practice, the technician is usually contacted by phone to inspect the issue and perform RM (Ch. 2.1.1 – Reactive Maintenance). According to interviews, operators have indicated that there is a mutual agreement with the technicians to inform them directly when a failure occurs. When communication occurs only between operators and technicians, supervisors and management may lack awareness of the current condition of production assets and their operational capabilities.

In addition, when maintenance technicians are unavailable or cannot be reached, information about failures may, in the worst cases, not be communicated to anyone. Alternatively, only a brief note describing the failure may be left on a sticky note for the next shift.

Occasionally, after the repair actions have been completed, the operator is present at the equipment and receives information about the completed repair directly from the technician. However, the relevant personnel may not always have the opportunity to remain at the equipment to observe the root cause or even to confirm whether the repair has been completed.

3.1.3 Reactive Maintenance Execution Process

Based on the author's observations, the technician begins fault diagnosis to identify the root cause of the failure. Once the cause has been determined, corrective actions are carried out (e.g., component replacement). After completing the corrective actions, the equipment's functionality is tested, and the process quality is evaluated. Following the repair, the maintenance technician returns to their office workstation to record a description of the event on the repair page of the equipment's information sheet.

In some cases, more urgent and higher-priority tasks may need to be completed before the failure is documented, which can result in missing or inaccurate repair information. A technician stated in an interview that resolving an equipment failure is prioritized over documenting it. As a result, it is not uncommon for certain failures and repairs to go unreported in the equipment information sheets.

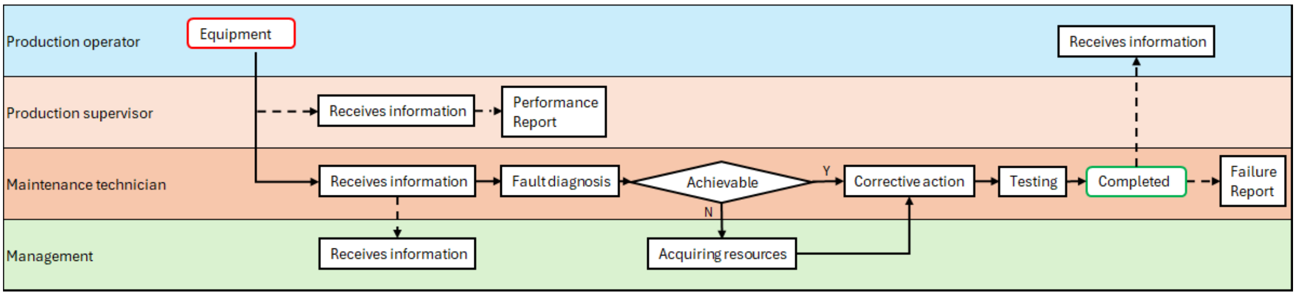


Figure 9 - The actual reactive maintenance process, as observed, is illustrated in the diagram above. Dashed lines indicate occasional information sharing.

3.1.4 Work Orders and Resource Management

According to a management interview, the client's maintenance technicians are also assigned method and quality development tasks in addition to their regular maintenance duties. They also work on project-related activities such as production tooling and layout design, meaning that task prioritization may vary throughout the day. However, there are no dedicated resource management or prioritization tools available for *"job-shop scheduling"* (Stevenson & Kull, 2024, p. 706), which makes it difficult to determine the most urgent tasks, especially when multiple stakeholders are involved simultaneously.

The client has previously attempted to use a follow-up list to support task prioritization, but according to the author's observations, it was discontinued because it did not include tasks from other departments. As a result, other departments' managers may not have been aware of the technicians' workload and ongoing tasks, which has led to situations where individual managerial priorities have been incorrectly treated as high-priority maintenance tasks. This highlights the need for a clear prioritization rule and for all maintenance-related tasks to be managed within a single system.

Production failures and PM tasks are typically given the highest priority. Other tasks are generally handled in coordination with management when necessary, under the assumption that technicians can independently manage their workload. However, according to the author's observations, this does not always reflect operational reality, as technicians may lack visibility of all scheduled tasks and therefore cannot effectively prioritize competing assignments.

According to a maintenance technician interview, change order projects initiated by other departments have in some cases been scheduled without ensuring that required prerequisites (e.g., components) are available. This has led to inefficiencies when technicians begin work only to discover that preceding workflow steps have not been completed, resulting in confusion and frustration.

3.1.5 Other Maintenance-Related Processes

According to the client's process description, the purchasing of manufacturing assets lacks defined responsibilities. When spare parts are required, the request for specific components is sent to a purchaser, who initiates the purchasing process in the ERP system. The components are then ordered. However, according to the process description and author's observations, after ordering there is no systematic follow-up for the components, and responsibility for replacement remains within the stakeholders. Purchase orders remain open in the ERP system until delivery, but there is no structured follow-up process for incoming spare parts.

According to an interview, the finance department can retrieve maintenance-related costs from the ERP system. However, the data must be manually processed, which means that preparing a cost report for a specific piece of equipment takes time, and that comparing multiple pieces of equipment takes even longer.

The client has a process for improvement initiatives, allowing any employee to propose ideas related to operational improvement. These initiatives are stored in individual system sheets and may relate to equipment usage or maintainability.

3.1.6 Maintenance Documentation

The client uses a manually updated system to document maintenance activities for production equipment. Each production asset has an equipment information sheet that records specifications, maintenance planning, and service history.

The equipment information sheet is divided into several sections, including a main page (equipment specifications and PM), instructional photos, daily inspections, repairs, set parameters, poka-yoke controls, drawings, a spare part list, a change log, and a 3D section.

The repairs section records completed RM actions, including responsible personnel, timestamps, and descriptions. According to the maintenance process description, maintenance technicians are required to complete a failure log including the date, time, cause of failure, repair time, and corrective actions taken. This information enables the calculation of MTBF and MTTR, which can be used to assess equipment availability and OEE. The failure log is maintained within the equipment information sheet.

The set parameters section defines operating limits and thresholds, while the poka-yoke section verifies error-prevention functionality in accordance with the control plan. The spare parts section lists spare parts for each piece of equipment on a subpage of the equipment information sheet. The IATF 16949 quality standard requires monitoring minimum stock levels for critical spare parts. The technical drawings section includes links to drawings, and the spare parts list includes part names, manufacturers, type numbers, suppliers, and stock values. The change log records updates, and the 3D section provides exploded assembly views with part numbering.

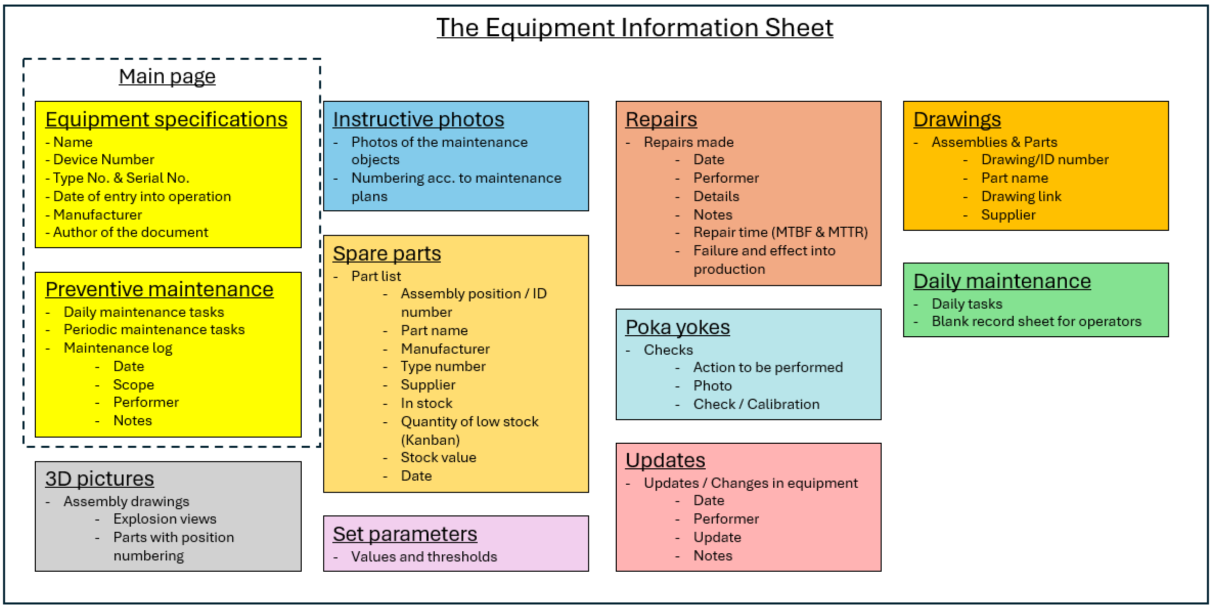


Figure 10 - Contents of the client's current equipment information sheet.

3.1.7 Regulatory Requirements

When new manufacturing tools or equipment are acquired, an equipment approval document must be completed. The document includes key information such as intended product use, device name, manufacturer, date, and approvers. It also functions as a checklist to verify that the equipment meets the required specifications and is approved for production use.

Directive 2006/42/EC (Official Journal of the European Union, 2019, p. 7; art 5 para 1[b]) and Finnish legislation (400/2008, 2011; 2004/1016, 2021) require the preparation of a "*technical file*" for machinery, if the equipment is classified as a "*machinery*" (Official Journal of the European Union, 2019, pp. 4-6; art 1 & 2).

Key aspects such as safety, maintainability, hazard identification and risk reduction through risk assessment, user instructions and training must be addressed before the machine can be put into production use (Occupational Safety and Health Administration in Finland, 2023; 403/2008, 2019). In Finland, both the manufacturer and the end user (employer) are responsible for machinery safety (Siirilä, 2008).

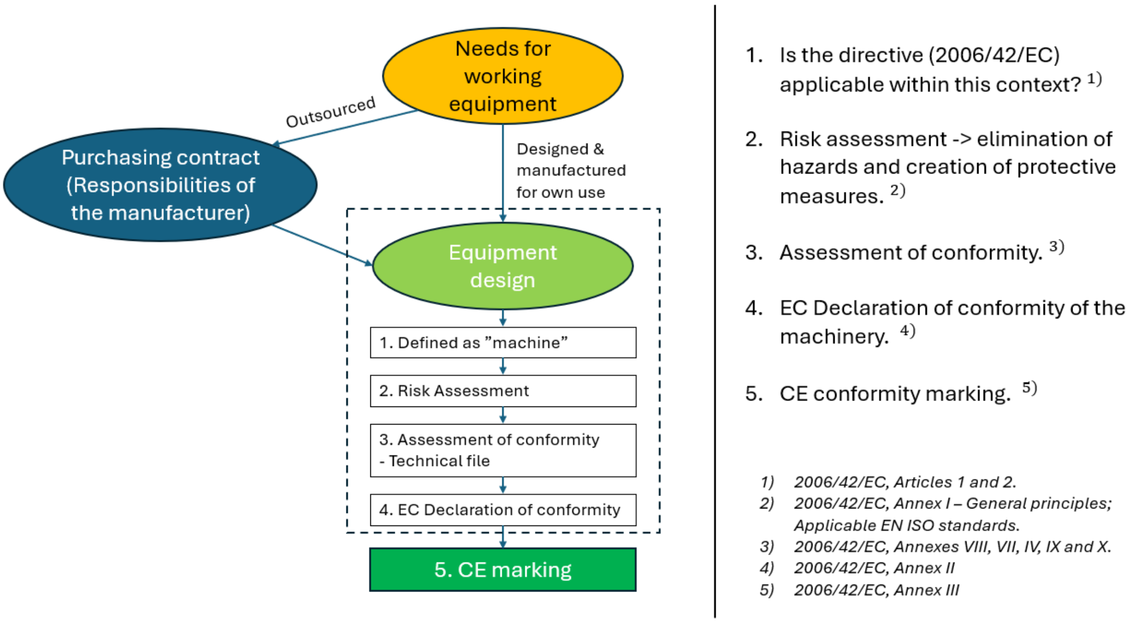


Figure 11 – Legal requirements for machinery set by Directive 2006/42/EC of the European Parliament and of the Council on machinery (METSTA, Metalliteollisuuden Standardisointiyhdistys ry, 2020, pp. 11 - 27).

3.2 Analyses

3.2.1 Analysis of the Preventive Maintenance Process

As previously described (Ch. 3.1.1 – Preventive Maintenance Process), daily inspections are recorded manually, which is not an efficient method, as the same information must later be scanned, transferred to the client's database, and copied into the equipment information sheet. This issue can be addressed by enabling reporting within the CMMS, making the records digital and available real-time.

The PM schedules are only available in the equipment information sheet, meaning that the maintenance technicians must check the inspections and maintenance contents from an office workstation. A CMMS accessible via mobile devices would allow the schedules and related instructions to be checked directly beside the equipment, without the need to return to the office. According to Järviö & Lehtiö (2012), it would also be beneficial to collect data on the time spent on PM tasks, as this would support maintenance planning (Järviö & Lehtiö, 2012, pp. 103 - 110).

The collection of data is not the only limitation in the follow-up process of equipment usage and maintenance, as monitoring is carried out separately for each piece of equipment. This means that there is no consolidated overview where assets can be compared side by side, with the "*Monthly Calendar of PM Tasks*" sheet being the only exception. Therefore, an overview page for equipment in the CMMS would be beneficial for monitoring the real-time status of assets.

3.2.2 Analysis of the Reactive Maintenance Communication Process

The observations regarding the maintenance communication process indicate that employees may perceive reporting as overly laborious. In addition, the personnel skill sets may be insufficient to fully understand the problem and the importance of reporting it, which is a noted issue (Järviö & Lehtiö, 2012, p. 81). Missing or incomplete information about the correct equipment, component, and failure has led to prolonged downtime, as it has been difficult for technicians to identify and resolve the exact issues. Therefore, it is essential to ensure that the required data – such as, time, informer, equipment ID, component, and description of the failure – is captured when reporting on equipment failures.

As the process description deviates significantly from actual communication practices, it would be beneficial to update it by implementing TPM strategies, in which production operators are trained and assigned responsibility for equipment maintenance and care. In addition, communication could be carried out on-site via mobile devices, requiring only essential information fields to be completed to simplify the reporting, thereby making communication easier.

The existing maintenance process for equipment failures contains gaps in the communication of equipment failure status information. As communication between operators and technicians leaves supervisors and management without sufficient situational awareness of the current state of production assets and their operational capabilities, task prioritization and resource allocation cannot be performed based on data.

This issue can be addressed by automating the process through system-generated notifications to relevant stakeholders whenever a failure occurs or its status changes. In addition, a continuously updated list of ongoing tasks is required to support effective prioritization and resource management.

3.2.3 Analysis of the Reactive Maintenance Execution Process

The author's observations show that technicians are able to perform RM at their own discretion when no formal prioritization method exists. As introduced in the previous chapter, transparent status information on ongoing tasks would form the basis for situational awareness, thereby enabling task prioritization. This highlights the need to establish a prioritization tool within the system that would automatically prioritize tasks based on predefined criteria.

The reporting of RM tasks and their status can be improved by using a mobile device-based system, where technicians can immediately report procedures on-site without having to return to the office workstation. This not only increases the availability of relevant data for decision-making but also improves reporting efficiency from a documentation perspective.

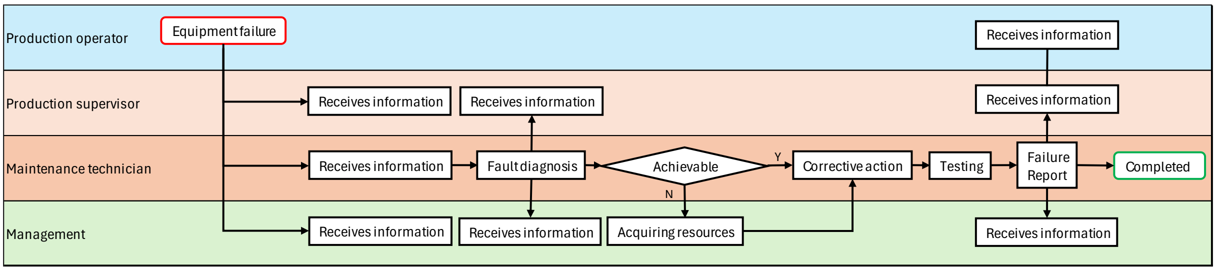


Figure 12 - Targeted reactive maintenance process after communication and execution analyses. Communication regarding maintenance actions should be automated so that all relevant stakeholders receive the information, thereby increasing process transparency and supporting resource allocation and task prioritization.

3.2.4 Analysis of Maintenance Documentation

As the investigation of maintenance documentation (Ch. 3.1.6 – Maintenance Documentation) shows, duplicate documents related to equipment management are stored in different locations. This creates inefficiencies in document retrieval and maintenance. Therefore, there is a need for a simplified and standardized data management approach, preferably based on a single master document linked to relevant files. For example, the equipment approval documentation (Ch. 3.1.7 – Regulatory Requirements) could be integrated into the equipment information sheet, since the technical file defines the requirements for maintenance operations when equipment is classified as machinery. The equipment information sheet could serve as the master document, which would improve accessibility and reduce redundant data storage and administrative workload.

The existing equipment management information is sufficient, but because it is maintained manually, it requires resources whenever changes occur. The amount of documentation is also increasing as new product lines are introduced, and duplicate information means that multiple documents must be updated when dependencies exist. For example, since some equipment shares the same components, maintaining the spare parts inventory within each equipment's information sheet requires excessive effort.

While the current equipment information sheet provides easy access to detailed spare parts data for individual equipment, it does not indicate spare part availability or how widely a specific part is used across different equipment, which limits its usefulness in supporting maintenance actions (e.g., when a technician needs to verify the availability of components required for an upcoming replacement during a PM task). In addition, standalone information sheets do not reflect real-time inventory levels. The previously identified shortcomings in the spare parts purchasing process (Ch. 3.1.5 – Other Maintenance-Related Processes) further emphasize the need for a centralized system to track incoming spare parts.

Therefore, there is a need for a centralized spare parts inventory database. The inventory should be maintained as a single master file linked to each piece of equipment, ensuring that updates (e.g., component replacements) are recorded only once. This need is also driven by quality requirements, as the standard requires monitoring minimum stock levels for critical spare parts.

According to the previously described manually driven cost reporting process used by the finance department (Ch. 3.1.5 – Other Maintenance-Related Processes), there is a need to consider implementing cost monitoring in the new CMMS system. This would enable the collection of costs not only from spare parts and labor related to repairs, but also from production losses (including scrap and availability losses), providing a more complete view of the total costs generated by equipment in use.

As the initiative process is managed through individual system sheets, it is difficult to trace earlier initiatives related to equipment, as the system does not categorize them. In theory, initiatives could be traced by ID, date, or proposer. However, in practice, this lack of structured categorization has led to situations where similar initiatives have been proposed multiple times without recognizing previous work. This highlights the need to document all future improvement initiatives related to manufacturing processes and equipment in the CMMS.

Furthermore, as the client intends to adopt the IATF 16949:2016 Automotive Quality Management System Standard, the implementation of a TPM system would be beneficial. Clause 8.5.1.5 of the IATF 16949:2016 standard defines the minimum requirements for production equipment maintenance systems. Performance indicators such as OEE, MTBF, MTTR, and the follow-up of PM tasks are presented as examples of documented maintenance objectives that should be monitored during management reviews (International Automotive Task Force, IATF, 2016).

As these requirements involve extensive data collection and monitoring, a manually driven process – where information is recorded and calculated at an office workstation for each piece of equipment – would not be economically viable in the long term. This necessitates the automation of data collection and system-based calculation functions, allowing maintenance resources to be redirected toward more value-adding tasks.

3.2.5 Actions Based on the Analyses

These outcomes highlight the need for a maintenance management system that can be easily used by operators and technicians via mobile devices, without requiring access to a PC for reporting activities.

These improvements should be considered complementary, where TPM supports operational discipline and data quality, while system automation enables real-time information flow and decision support.

Based on observations related to resource management (Ch. 3.1.4 – Work Orders and Resource Management), there is a need for a CMMS-based work order system that integrates all maintenance-related tasks into a single platform. Prioritization should be clearly defined, and scheduling should be based on resource availability.

When all necessary equipment data is available, it can be sorted and filtered by equipment category to form a "*benchmarking*" tool for analysis (Järviö & Lehtiö, 2012, pp. 63 - 64 & 139 - 140). For example, equipment with the highest number of failures can be presented in a "*Pareto analysis*" (Stevenson & Kull, 2024, p. 409), which can be used to prioritize maintenance activities and asset management decisions. By collecting data from daily operations, the required KPIs can also be measured.

An efficient, integrated computerized maintenance management system, which enables seamless data transfer between organizational levels (including management), would create additional value. It can be defined as mutual service between the production operators and maintenance. (Järviö & Lehtiö, 2012, pp. 155 - 156 & 195.)

4 Maintenance System Development – Ideal Model

4.1 System Design Framework

The AD method will be used to develop the new pilot CMMS. In the following chapter, the stakeholder needs and existing "*constraints*" (Cs) of the system will be identified, and the "*functional requirements*" (FRs) and corresponding "*design parameters*" (DPs) will be defined and mapped. "*Process variables*" (PVs) for the DPs will then be selected based on the capabilities of the CMMS. The design will also be evaluated, with the aim of achieving an ideal CMMS design. Together, this process will result in a system architecture for the new CMMS.

4.1.1 Customer Attributes (CAs) - Stakeholder Needs for CMMS

As introduced earlier (Ch. 2.2.1 – Axiomatic Design Theory), the design process begins by identifying stakeholders and mapping CAs. Based on the background research (Ch. 2 - Theory and Literature Review and Ch. 3 - Methods and Datasets), several stakeholder groups and their needs were identified. These groups can be categorized into client's existing departments, which define different levels of informational needs, such as informing, reporting, monitoring, and benchmarking.

The stakeholders of the client can be identified as follows:

1. Maintenance technicians (C1).
2. Production operators (C2).
3. Management (C3-C7), including maintenance supervision, quality management, finance management, operations management and production planning.

The client's maintenance technicians (C1) perform a wide range of tasks related to production equipment, including PM, spare parts inventory management, handling equipment breakdowns, and documenting maintenance activities in equipment logs. Based on the analysis, the technicians' needs include at least the following:

- As previously indicated (Ch. 3.2.1 – Analysis of the Preventive Maintenance Process), maintenance technicians should be able to access and review PM tasks regardless of their location (CA11).

- According to the analysis of spare part inventory (Ch. 3.2.4 – Analysis of Maintenance Documentation), technicians should be able to view available spare parts and manage the spare parts inventory (CA12).
- As suggested by the analysis (Ch. 3.2.2 – Analysis of the Reactive Maintenance Communication Process), the process of documenting equipment failures should be simplified (CA13), including:
    - Assigning the failure to the correct tool, device, or machine.
    - Automatic recording of the start date and time.
    - The ability to describe the failure.
    - The ability to document corrective actions performed.
    - The ability to select appropriate spare parts with visibility into inventory levels and associated costs.
    - Automatic recording of the completion time (temporary or permanent resolution).

The production operators (C2) are responsible for documenting their daily inspections and identifying deviations from specified and normal operation conditions. In addition, the need to improve communication and information quality related to equipment failures requires a simpler and more efficient reporting method for operators to inform relevant stakeholders. The operators' needs have been identified as follows:

- As determined in the analysis (Ch. 3.2.1 – Analysis of the Preventive Maintenance Process), the reporting of daily inspections should be simplified (CA21):
    - Equipment and environmental conditions must meet specified requirements.
    - Nonconforming conditions should be reported and forwarded to management.
- As previously represented (Ch. 3.2.2 – Analysis of the Reactive Maintenance Communication Process), there should be an improved method for communicating and reporting equipment failures (CA22). The report should include:
    - Event number.

  - Identification of the correct equipment.
  - Accurate and concise description of the issue.
  - Routing of the report to the correct recipients.
  - Automatic recording of the event date and time.
- As identified in the analysis of maintenance documentation (Ch. 3.2.4 – Analysis of Maintenance Documentation), there is a need to improve equipment-related documentation, enabling operators to submit improvement initiatives related to production processes and equipment (CA23).

The management requires more precise data regarding production, production planning, and equipment. Maintenance, quality, finance, and operations management all require accurate data as a basis for decision-making. According to the analyses (Ch. 3.2 – Analyses), the identified needs are:

- The maintenance supervisor (C3) needs access to resource usage data (CA31).
- The maintenance supervisor (C3) needs the ability to prioritize tasks (CA32).
- The maintenance supervisor (C3) needs the ability to identify the most critical failures (CA33).
- The maintenance supervisor (C3), together with the production planner (C7), needs the ability to schedule maintenance tasks, for example in relation to production shutdown planning (CA34).
- Quality management (C4) needs reports when new continuous improvement initiatives are introduced (CA41).
- Finance management (C5) needs the ability to perform cost tracking to support capital expenditure and investment planning (CA51).
- Operations management (C6) needs access to equipment failure data, particularly when production incentives are based on productivity measures (CA61).

- Operations management (C6) needs to monitor TPM-related KPIs, such as MTBF, MTTR, and OEE, and to support employee engagement through data-driven incentives (CA62).
- The production planner (C7) needs the ability to see equipment unavailability due to failures (CA71).

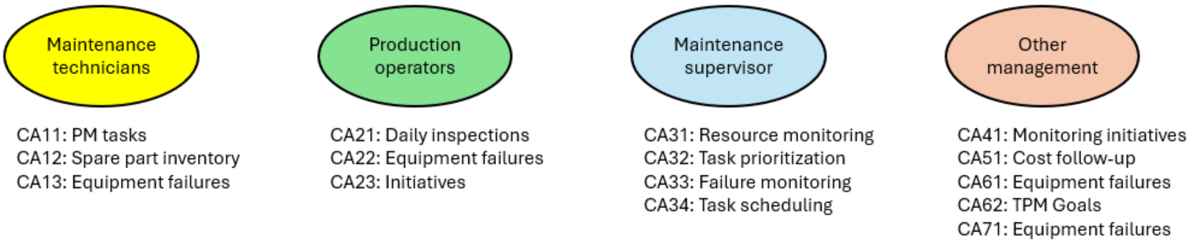


Figure 13 - Stakeholders and Customer Attributes.

As shown in Figure 13, various needs have been identified among the client's stakeholders related to maintenance activities. As the first research question addresses the scope of the design, there is a need to identify, structure, and prioritize the information that the new CMMS should capture. Some stakeholder groups share similar needs, and these will be prioritized, as they define the boundaries of the CMMS development. However, the needs excluded from this study will serve as input for future development phases.

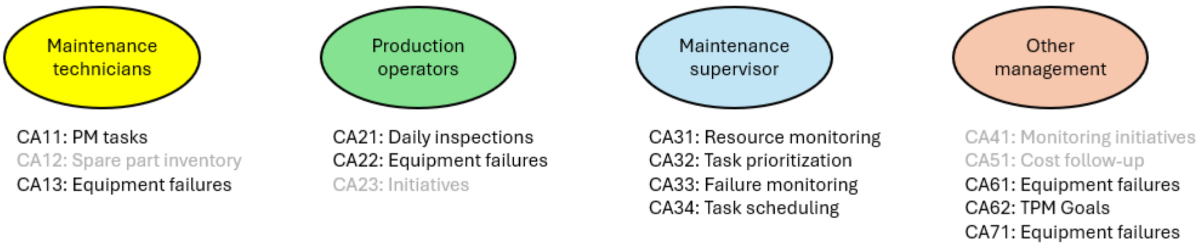


Figure 14 – Chosen CAs for the initial System Design (highlighted in black text).

4.1.2 Functional Requirements (FRs) and Constraints (Cs)

The FRs can be derived once the stakeholders and their needs have been identified (Ch. 2.2.1 – Axiomatic Design Theory). The objective is to develop a CMMS for managing the maintenance activities; therefore, this is defined as FR_0.

FR_0 = Create Computerized Maintenance Management System
Cs = No-code platform

Figure 15 - The mapping process starts from top tier, indicated as FR_0.

The stakeholder-dependent attributes generate requirements for the CMMS to manage PM tasks (FR_1), equipment failures (FR_2), and resources (FR_3).

4.1.3 Design Parameters (DPs)

The defined FRs can be satisfied by corresponding DPs, preferably in accordance with the previously introduced independence axiom (Ch. 2.2.2 – Design Matrix). Thus, each DP should be designed so that it does not affect other FRs within the system.

Since the study is constrained to use a no-code platform for the CMMS, the solution will be implemented through application-based functions for data collection and integration. The management of PM tasks (FR_1) can be addressed through a PM task application (DP_1). FR_2, related to equipment failures, will be handled through an application (DP_2) that enables an information flow to the relevant stakeholders. FR_3 requires resource management and is therefore addressed through a scheduling and prioritization application (DP_3) within the system.

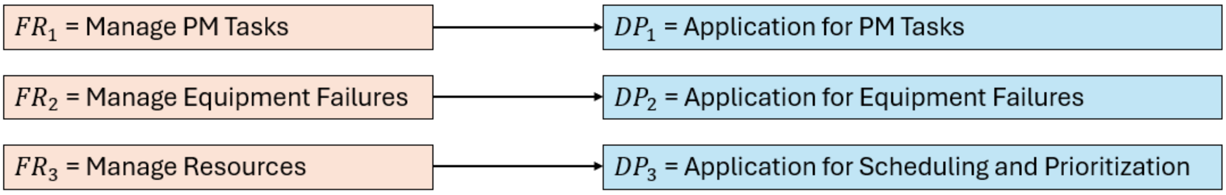


Figure 16 - The designer defines the first-level FRs and maps them to DPs.

4.1.4 Decomposition of FRs and DPs

The decomposition of the design starts from the previously defined higher-level requirements. Design parameter, DP_1 – *Application for PM Tasks*, is decomposed from the user perspective. PM tasks are performed daily, weekly, monthly, semi-annually, or annually, depending on the user role. More demanding and time-taking tasks are carried out by maintenance technicians based on the equipment maintenance program. Daily inspections and less skill-intensive maintenance and cleaning tasks can be assigned to operators, who should be encouraged to take responsibility for equipment, in line with TPM principles (Ch. 2.1.5 – Total Productive Maintenance, TPM).

It is important to define clear boundaries for the data to be collected so that the process adds value in the long term. Operators should only be able to select from predefined options, ensuring clear traceability. The process should be efficient, without excessive time consumption, and should avoid generating non-value-adding information. The application should collect only the essential information, similar to the previously described equipment information sheet (Ch. 3.1.6 – Maintenance Documentation), where date, scope, performer and notes were recorded.

To ensure sufficient traceability for each event reported in the system, the following information must be captured:

- Date and time.
- Equipment number.
- User.
- Tasks and checks performed.

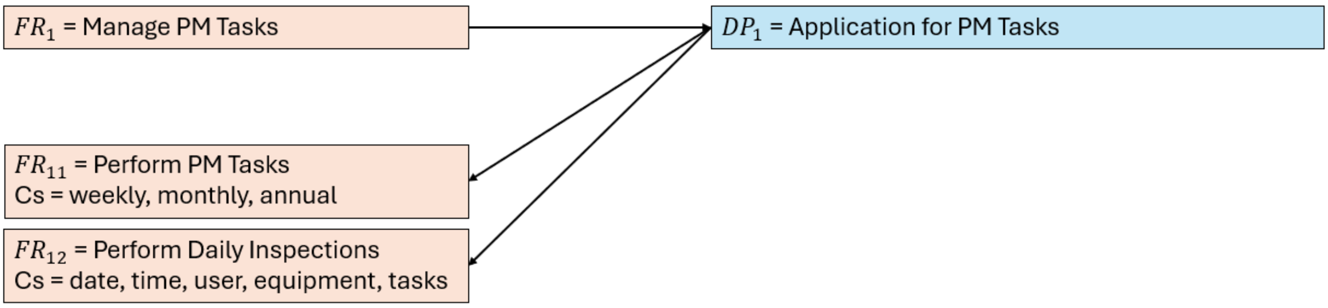


Figure 17 - Decomposition of DP_1.

The second design parameter, DP_2 – *Application for Equipment Failures*, is decomposed into functionalities based on the identified needs of the equipment failure workflow. As previously described (Ch. 3.2.2 – Analysis of the Reactive Maintenance Communication Process), communication and reporting of equipment failures have caused challenges. The identified CA22 related to failure reporting can be addressed through an application for managing the equipment failures (DP_2).

The first requirement of the application is to alert and forward the information about equipment failures; thus, the first decomposed functional requirement is FR_{21} – *Inform about Failures*. Equipment failures require traceability in the same way as PM tasks; therefore, FR_{21} must capture the answers to questions: when, who, where, and what.

FR_{22} – *Perform Corrective Actions* is another requirement of the failure workflow and is used to record corrective action information. FR_{22} also ensures traceability of the repair process;

therefore, it should include the date, time, technician, equipment, failure location, and description of corrective actions.

FR_{23} – *Follow Up Failures* is required at the management level to monitor equipment availability and to analyze costs caused by failures.

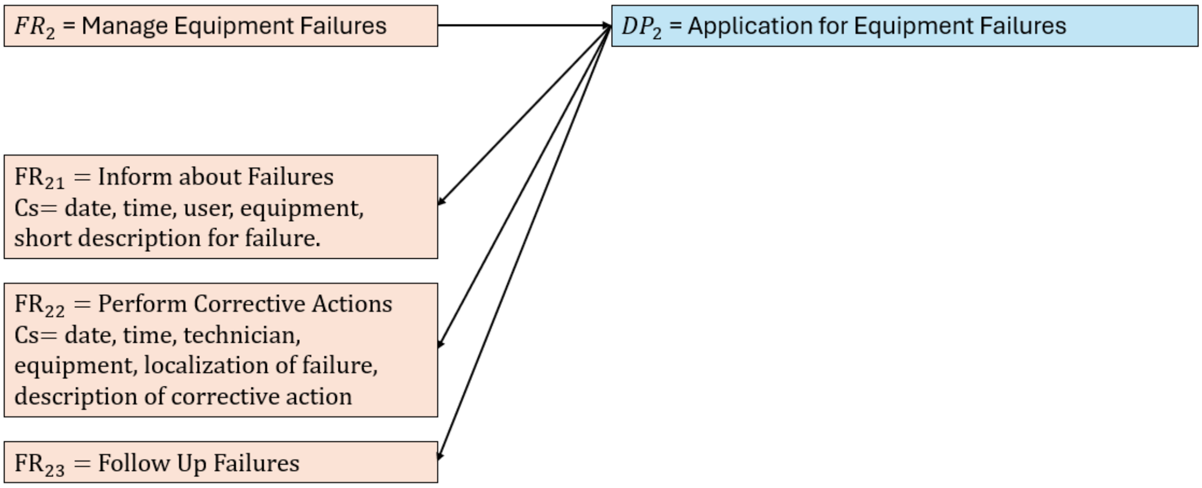


Figure 18 - Decomposition of DP_2.

The third design parameter, DP_3 – *Application for Scheduling and Prioritization*, is decomposed to support the work of maintenance technicians. FR_{31} – *Prioritize and Schedule Tasks* is required to improve maintenance resource management, and FR_{32} – *Create Work and Change Orders* is required from a stakeholder perspective to enable access to maintenance resources.

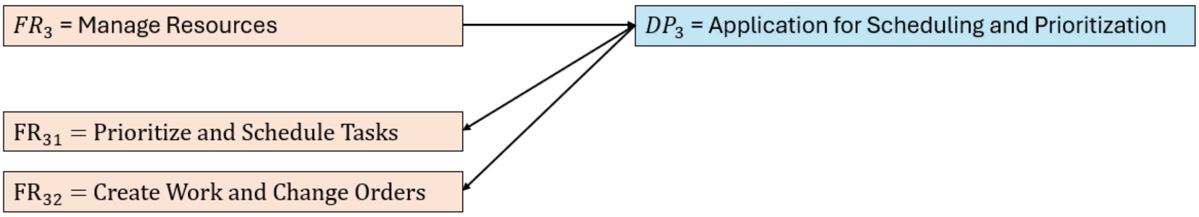


Figure 19 - Decomposition of DP_3.

4.1.5 Lower-Level DPs

After defining the top level of the design, it is further decomposed into lower-level FRs and DPs to provide a more detailed view of the system architecture. Since the aim is to develop the system using a no-code platform, different application modules will be used to fulfil the FRs.

To perform weekly, monthly, or annual PM tasks (FR_{11}), DP_{11} – *Maintenance Technician Module for PM Tasks* will be designed. Another module, DP_{12} – *Operator Module for Daily PM Tasks*, will be used by production operators to perform daily inspections (FR_{12}).

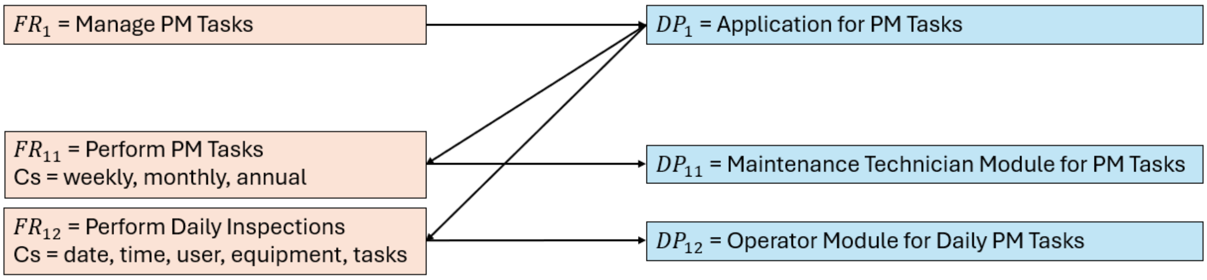


Figure 20 - PM tasks will be performed either by the operator or by the maintenance technician.

FR_{21}, which requires informing about failures, is satisfied by DP_{21} – *Reporting Module*. To record corrective actions (FR_{22}), DP_{22} – *Maintenance Technician Module for Corrective Actions* will be used. To follow up failures (FR_{23}), DP_{23} – *Management Module* will be created.

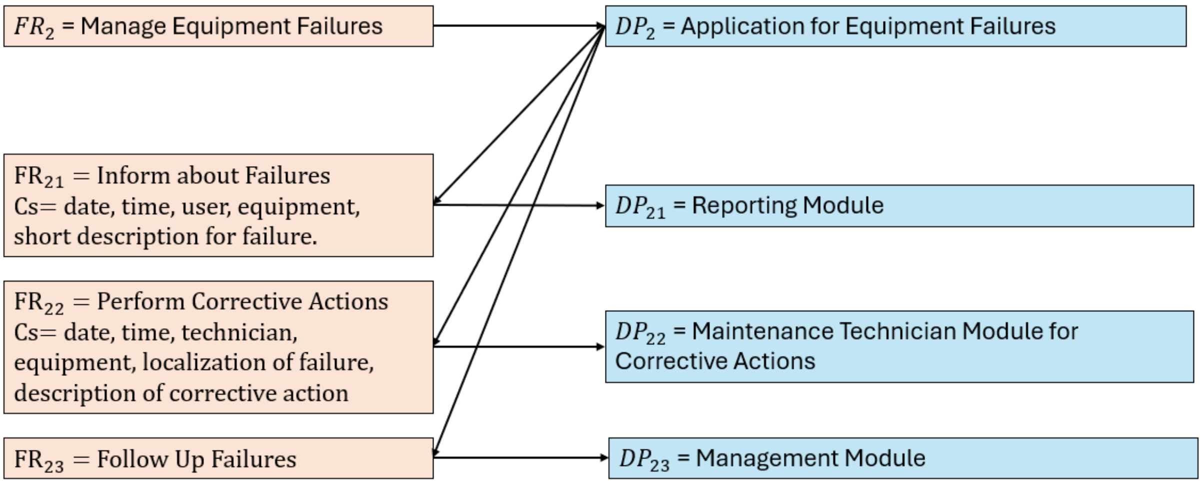


Figure 21 - Design modules for handling equipment failures.

The resource management will have separate modules for prioritization and scheduling of tasks (FR_{31}) and for work and change order creation (FR_{32}). The first will be fulfilled by using DP_{31} – *Maintenance Supervision Module*, and the second using DP_{32} – *Work Queue Module*.

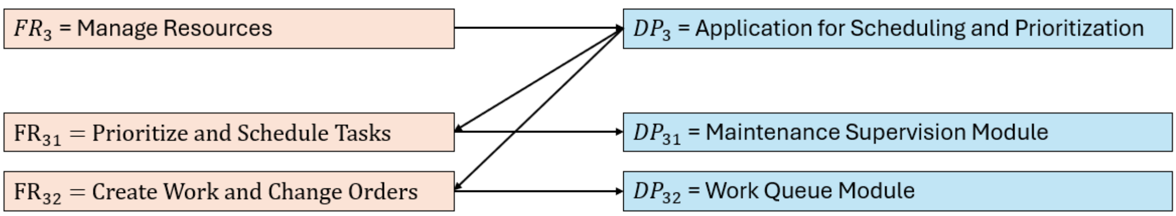


Figure 22 - Modules for resource management.

4.1.6 Application for PM Tasks (DP1)

The maintenance technician module will be designed to fulfil the PM program. Each piece of equipment has its own program, the contents of which are available in the equipment information sheet. To complete PM tasks correctly, the technician must ensure that the correct program and tasks are executed.

To ensure proper system functionality, the module must first allow selection of the correct equipment and subsequently the appropriate maintenance program based on the date and service history. Service history must be accurately recorded to provide sufficient data on task completion, PM costs, and equipment availability. Time usage and service intervals are important for future cost calculations.

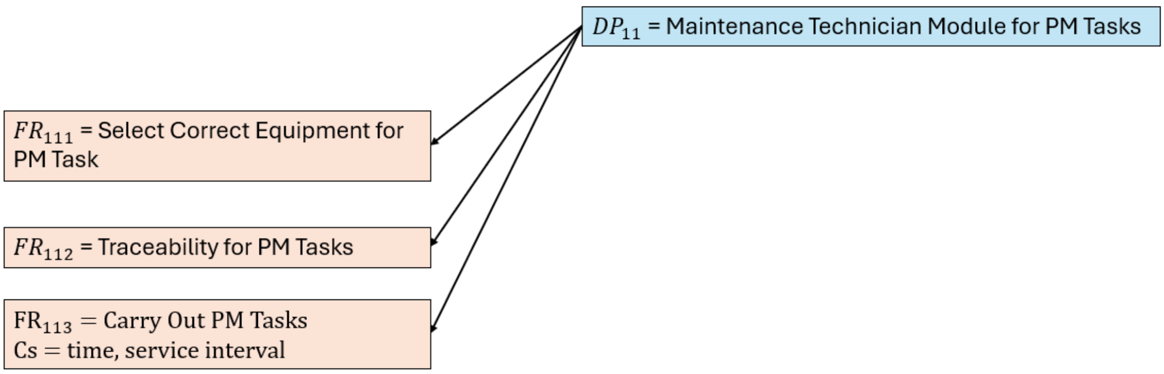


Figure 23 - Maintenance Technician Module for PM Tasks and Its FRs.

FR_{111} is the first functional requirement, defining the selection of equipment. There are several ways to identify the correct equipment for PM tasks. To date, technicians have selected PM tasks from the equipment information sheet, and the correct equipment file has been chosen from a list of equipment located in the monthly calendar (Ch. 3.1.1 – Preventive Maintenance Process). Thus, using an existing list from which to select equipment is one possible concept.

The aim of the new CMMS is to be mobile; therefore, it is beneficial to utilize mobile device functionalities for equipment identification. One concept is to use a camera with a QR code to identify the equipment. Another concept is voice recording, allowing the technician to dictate the equipment number into the system. The final concept is a free-text field where the equipment name or number can be entered manually.

One of the four concepts will be selected. The selection will be based on a criteria table in which the different options are evaluated, compared, and ranked.

	Concept variants					
	A	B	C	D		
Preventing human error	4	3	2	1		A = QR code
Efficiency	4	2	3	1		B = Equipment list
Dependency (on equipment & program)	2	3	1	4		C = Voice recorder
Updateability / Programming	2	3	1	4		D = Free writing
Mobility	4	2	3	1		
Feasibility	2	3	1	4		
Total points	**18**	16	11	15		

Table 5 - Concept selection table used to choose the equipment recognition method for the technicians.

The QR code received the highest rating in the concept comparison for equipment recognition (i.e., providing more value than alternative concepts) and was selected as DP_{111}. The criterion "prevention of human error" reflects the risk of users confusing different pieces of equipment (e.g., selecting the wrong equipment ID from a list or entering incorrect information in a text field). Efficiency evaluates the time required to perform the action. Dependency refers to external factors that may affect the concept (e.g., a worn or damaged QR code label). Updateability and programming describe the effort required to modify or maintain the application module. Mobility measures the flexibility provided by the concept. Feasibility reflects the overall resources required for implementation.

FR_{112} – *Traceability for PM Tasks* requires information about the performer. Four different concepts were developed for user identification: RFID-based identification, a user list, voice input, and free-text input. Based on the selection criteria table, RFID was selected as the solution, defined as DP_{112}:

	Concept variants					
	A	B	C	D		
Preventing human error	4	3	2	1		A = RFID recognition
Efficiency	4	2	3	1		B = User list
Dependency (on equipment & program)	2	3	1	4		C = Voice recorder
Updateability / Programming	2	3	1	4		D = Free writing
Mobility	4	2	3	1		
Feasibility	2	3	1	4		
Total points	**18**	16	11	15		

Table 6 - Concept variants for user identification, evaluated using a selection criteria table to determine DP_{112}.

The maintenance technician module for PM tasks requires information on both completed and uncompleted PM tasks and inspections. FR_{113} – *Carry Out Tasks* can be fulfilled through several design concepts.

One concept is an automatic setup that presents the required tasks in the correct order and requires the user to confirm whether each task has been completed. Another concept is a task list from which the technician selects the tasks performed. A third concept is a hybrid solution, combining a predefined task list that guides the technician, with additional documentation of completed tasks and equipment condition via voice input. The fourth concept is free-text input, which reflects the current practice, where the technician records completed tasks and equipment condition manually. The key difference compared to the existing method is that mobile access allows documentation to be performed directly at the equipment.

DP_{113} – *The Automatic Setup* provides the highest value, as it reduces working time and guides the technician based on service intervals, equipment usage, and date. It also limits the generation of non-value-adding data, whereas free-text and voice input approaches may produce unstructured information, making data filtering and analysis more difficult. The task list concept presents a higher risk of human error, as it relies on the technician selecting the correct tasks.

The main disadvantage of the automatic setup is the implementation effort, as it requires defining task-specific rules for different equipment and service intervals.

	Concept variants			
	A	B	C	D
Preventing human error	4	3	2	1
Efficiency	4	2	3	1
Dependency (on equipment & program)	2	3	1	4
Updateability / Programming	2	3	1	4
Mobility	3	2	4	1
Feasibility	2	3	1	4
Total points	**17**	16	12	15

A = Automatic setup
B = Task list
C = Voice recorder
D = Free writing

Table 7 - Automatic setup selected as DP_{113}:.

All design parameters for the Maintenance Technician Module have been defined. The module is illustrated in the figure below.

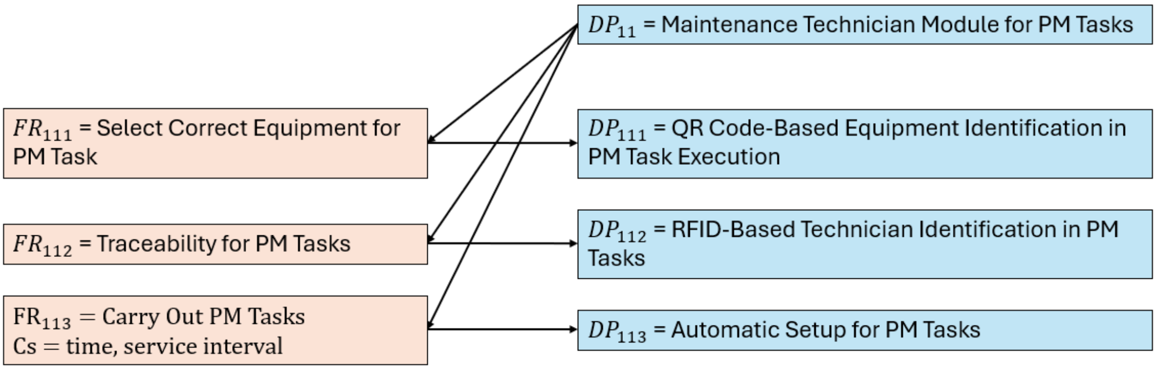


Figure 24 - DP_{11} - Maintenance Technician Module for PM Tasks, decomposed.

DP_{12} – *The Operator Module for PM Tasks* will be designed to fulfil the operator-driven PM program (i.e., daily equipment inspections). As previously presented (Ch. 2.1.5 – Total Productive Maintenance, TPM), operators can perform predefined equipment checks and simple maintenance tasks. These tasks require daily follow-up, and the Operator Module shares largely the same requirements as the Maintenance Technician Module (Ch. 4.1.6 – Application for PM Tasks). The main difference between the modules lies in the scope and skill level required, as operator tasks and inspections are less demanding than those performed by maintenance technicians.

The Operator Module has the following functional requirements:

- FR_{121}– *Select Correct Equipment for Daily Inspections*
- FR_{122} – *Traceability for Daily Inspections*
- FR_{123} – *Carry Out Daily Inspections*

There is no need for a separate concept comparison, as these functional requirements have the same objectives as those of the maintenance technician module (i.e., equipment selection and task traceability). FR_{121} and FR_{122} can be fulfilled using similar design parameters as defined earlier: DP_{121} – *QR Code-Based Equipment Identification in Daily Inspections* and DP_{122} – *RFID-Based Operator Identification in Daily Inspections.*

FR_{123} will also be implemented using an automatic setup; however, it is specifically designed for the operator module and is therefore defined as DP_{123} – *Automatic Setup for Daily Inspections*.

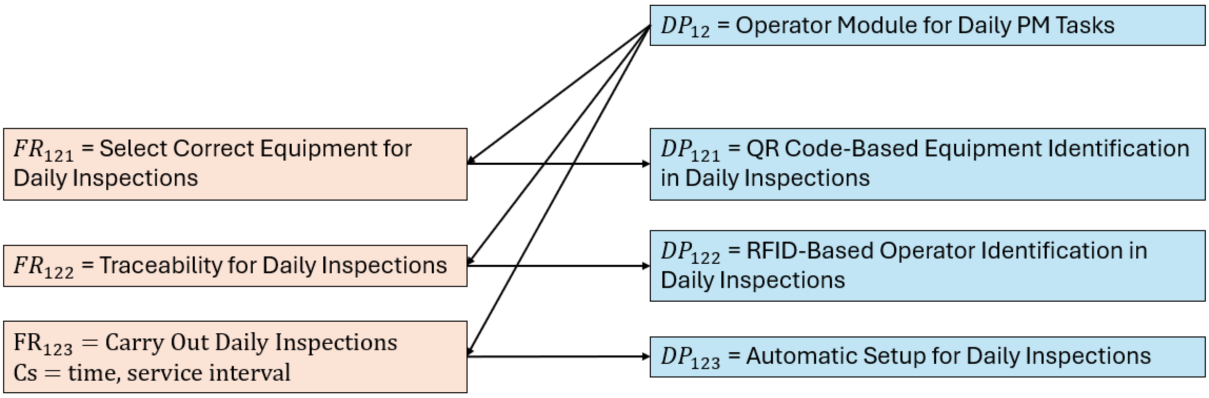


Figure 25 - DP_{12} - Operator Module for Daily PM Tasks, decomposed.

4.1.7 Application for Equipment Failures (DP2)

As discussed earlier (Ch. 3.2.2 – Analysis of the Reactive Maintenance Communication Process), the client has a clear need for a CMMS, particularly to improve the communication and follow-up of equipment failures. Currently, daily communication is challenging, and operators may feel that their failure reports are not adequately addressed. There is a lack of visibility into ongoing maintenance tasks, limiting situational awareness for employees.

Additionally, there is no structured prioritization of maintenance tasks, which can create unfavorable biases among operators regarding the maintenance department. Therefore, there is a need for a more efficient way to report failures and, importantly, to describe the issues in detail – identifying the equipment, the location of the problem, and the relevant contacts for further information.

Management also requires accurate, up-to-date information on the condition of manufacturing assets to make informed decisions. All decisions should be based on reliable and sufficient data to ensure the correct response to maintenance needs.

The Figure 2 (p. 13) illustrates a timeline for an ideal RM process within a maintenance management system. This timeline provides an overview of the key actions required throughout the repair process. To respond effectively to equipment failures, relevant information must be systematically recorded to support subsequent decision-making.

The process begins when an operator detects a failure or deviation from normal equipment operation and reports it to the maintenance department. The time of the incident should be recorded to enable tracking of MTBF (Ch. 2.1.6 – Mean Time Between Failures, MTBF). Upon receiving the report, the maintenance team must assess the situation and determine whether immediate action is required or if the diagnosis can be postponed.

Effective decision-making requires sufficient and accurate information. By categorizing the severity of failures, maintenance personnel can prioritize tasks within the work queue. The application used for scheduling and prioritization is discussed in more detail in the next chapter (Ch. 4.1.8 – Application for Scheduling and Prioritization).

After receiving the failure report, the maintenance technician begins diagnosing the issue. The time spent on diagnosis and repair must be documented to enable the calculation of MTTR (Ch. 2.1.7 – Mean Time to Repair, MTTR). Corrective actions taken to repair the equipment (e.g., replacement of a faulty component) should also be recorded for further analysis and evaluation.

The reporting of corrective actions should be straightforward for the technician. Additionally, reports on failures and their repairs should be presented in an analyzable format for management. All machines should be compiled into lists and visualized through charts based on the values of the previously introduced KPIs (Ch. 2.1.6 – Mean Time Between Failures, MTBF; 2.1.7 – Mean Time to Repair, MTTR; 2.1.8 – Overall Equipment Effectiveness, OEE).

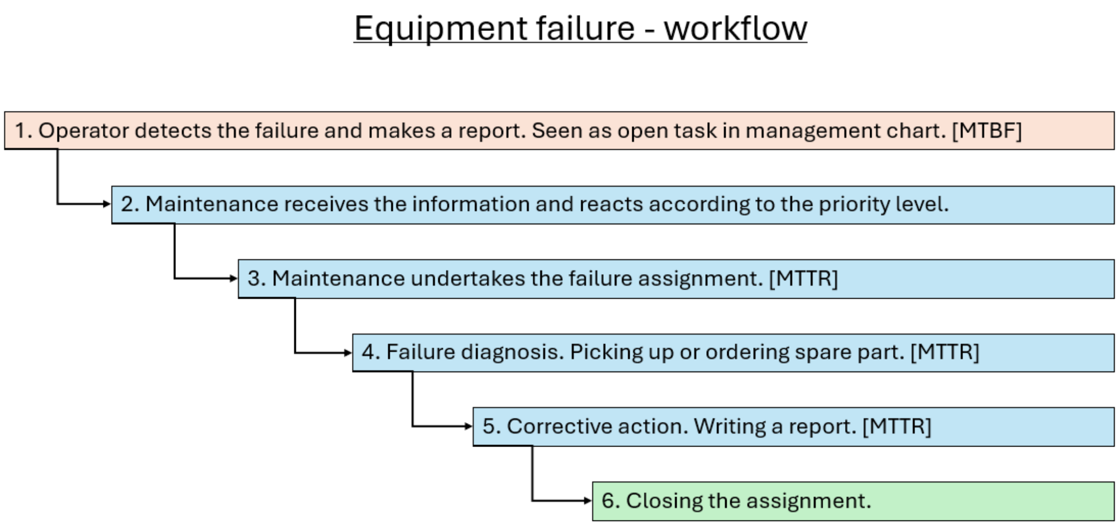


Figure 26 - Equipment failure workflow.

In failure reporting, the operator must indicate the correct equipment where the failure has occurred (FR_{211} – *Select Equipment under a Failure*). To support this, DP_{211} – *QR Code-Based Equipment Identification in Failure Reporting* will be implemented. In addition, the identity of the failure reporter must be recorded in the failure log (FR_{212} – *Traceability for an Equipment Failure*). For this purpose, DP_{212} – *RFID-Based Operator Identification in Failure Reporting* will be used, as RFID tags enable personal user identification.

The reporting system also includes a third requirement (FR_{213} – *Inform about a Failure*), which ensures that relevant information is delivered to the appropriate stakeholders with an appropriate level of detail. To address this, DP_{213} – *Automatic Information Setup to Report about a Failure* will be designed.

These three functionalities form the basis for the DP_{21} – *Reporting Module*.

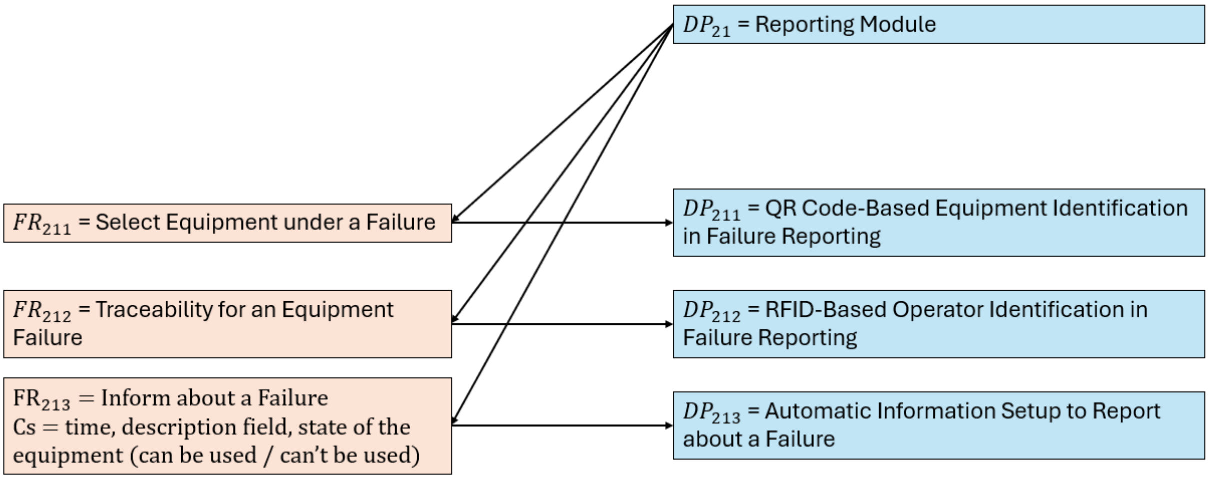


Figure 27 - DP_{21} - Reporting Module.

After a failure has been reported, the information is delivered to the maintenance personnel, who address it within the DP_{22} – *Maintenance Technician Module for Corrective Actions*. According to the previously described workflow in Figure 26, this module is defined by four requirements:

- FR_{221} – *Get Information about a Failure*
- FR_{222} – *Select Equipment to Start Corrective Action*
- FR_{223} – *Undertake Assignment*
- FR_{224} – *Diagnosis / Localization of Failure and Documentation of Corrective Actions*

DP_{221} – *Automatic Notification about a Failure from the Information* Setup ensures that maintenance personnel receive relevant failure information, fulfilling the FR_{221}. DP_{222} – *QR Code-Based Equipment Identification under a Failure* enables the technician to access detailed information regarding the affected equipment and the ongoing failure, fulfilling FR_{222}. DP_{223} – *RFID-Based Technician Identification for Corrective Actions* enables access to execute the task and indicates that the corrective action is in progress, fulfilling FR_{223}.

After diagnosis and corrective actions have been completed (FR_{224}), the results are documented in the CMMS through DP_{224} – *Failure Management Setup*.

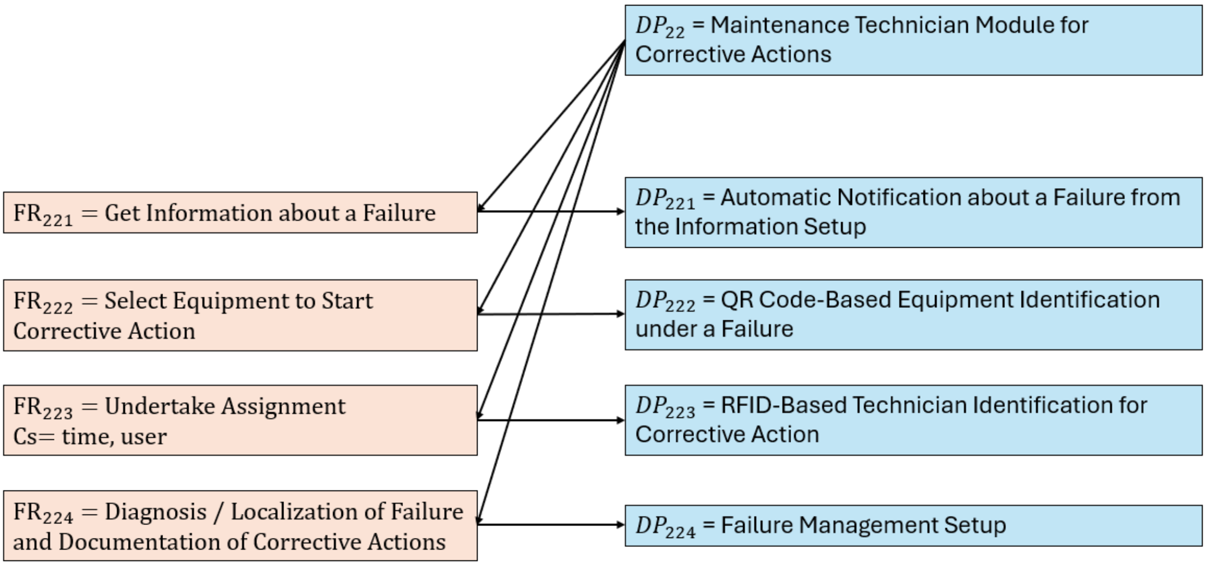


Figure 28 - DP_{22} - Maintenance Technician Module for Corrective Actions.

The third module, DP_{23} – *Management Module*, is used to deliver information to the management team. FR_{231} – *Status of PM Tasks* is the first requirement, and this information will be shared through DP_{231} – *Non-Conformance Report for Equipment*. There is also a need for management to view current failure statistics, leading to FR_{232} – *Access to Equipment Failure Data*. These failure statistics will be presented using DP_{232} – *Chart of Equipment Failure Statistics Based on KPIs*.

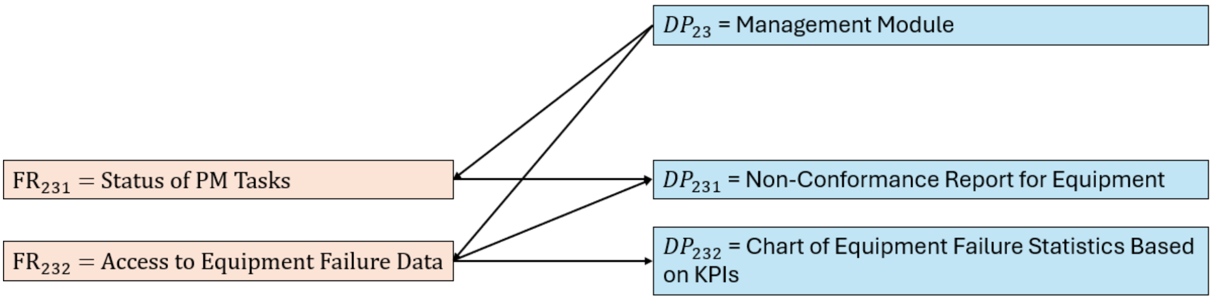


Figure 29 - DP_{23} - Management Module.

4.1.8 Application for Scheduling and Prioritization (DP3)

The third application will be developed for resource management. DP_{31} – Maintenance Supervision Module is used to control tasks assigned to maintenance. As failures are reported, they must be prioritized and scheduled based on their risk level, leading to FR_{311} – *Set Priority and Schedule for RM Tasks*.

The risk level will be defined using DP_{311} – *Automatic Prioritization of RM Tasks (FMEA)*, where safe equipment operation is ensured and the economic consequences for operations, as well as the impact on finished goods, are evaluated (see more on FMEA in Ch. 2.1.4 – Reliability-Centered Maintenance, RCM).

There is also a need to adjust prioritization manually, which defines FR_{312} – *Change Prioritization and Scheduling for Maintenance Tasks* as the second functional requirement of the module. DP_{312} – *Admin Tool for Work Queue* will be developed to fulfil FR_{312}, enabling authorized users to override the work queue regardless of assignment priority levels.

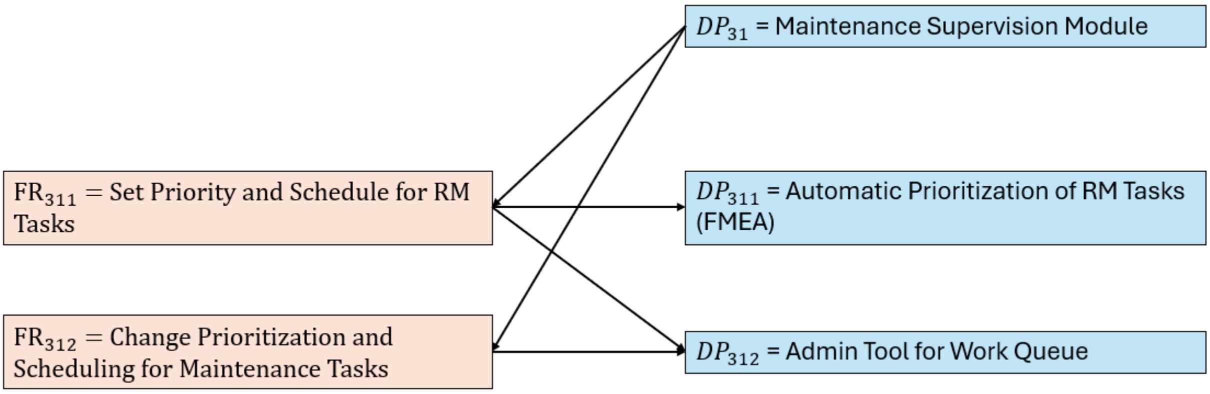


Figure 30 - DP_{31} - Maintenance Supervision Module for Prioritization and Scheduling of Maintenance Tasks.

The DP_{32} – *Work Queue Module* enables other departments to create work orders to be executed by the maintenance department. These work orders may include improvement requests related to equipment usability, maintainability, or any other tasks assigned to maintenance. The module consists of five functional requirements to ensure clear traceability of work orders. It is essential that the correct target of each order is identified and that the system verifies the authorization of the user submitting the request, including budget-related constraints.

The FRs are:

- FR_{321} – *Check User Rights*
- FR_{322} – *Create Work Order*
- FR_{323} – *Indicate Budget and Work Hour Estimation [Cs: €, h, investment no.]*
- FR_{324} – *View Work Orders*
- FR_{325} – *Manage Work Order*

DP_{321} – *RFID-Based Identification for Work Order* will be implemented to verify that the user has the rights to submit or edit a work order (FR_{321}). DP_{322} – *QR Code-Based Equipment Identification for Work Order* will be used during work order creation to ensure the correct equipment is linked to the task (FR_{322}). DP_{323} – Cost Calculation Tool will be used for budget and work hour estimation (FR_{323}). This tool allows maintenance to evaluate the scope of the assignment and plan the execution of the task according to the allocated budget. If the estimated costs exceed the approved budget, the maintenance supervisor can either reject the order or return it to the requester for re-evaluation. This is where *Create Work and Change Orders* (FR_{32}) is influenced by DP_{31} – *Maintenance Supervision Module*. FR_{323} has three constraints: cost [€], estimated hours [h], and investment number [#].

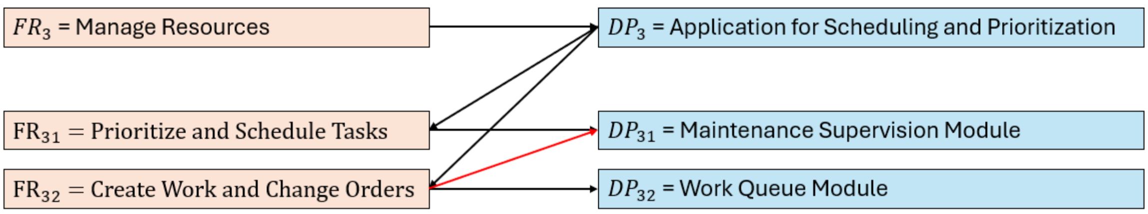


Figure 31 - Deviation from Figure 22 – DP_{31} Maintenance Supervision Module influencing FR_{32} Create Work and Change Orders

As work orders are created, they are listed chronologically and according to their priority in DP_{324} – Work Queue List to satisfy the requirement of "*Prioritize Work Orders*" (FR_{324}). If higher-priority tasks arise, the *Maintenance Supervision Module* (DP_{31}) allows manual reprioritization, supporting a holistic approach to maintenance resource management. To execute a work order (FR_{325}), DP_{325} – *RFID-Based Technician Identification for Work Order* is used to register the technician performing the task.

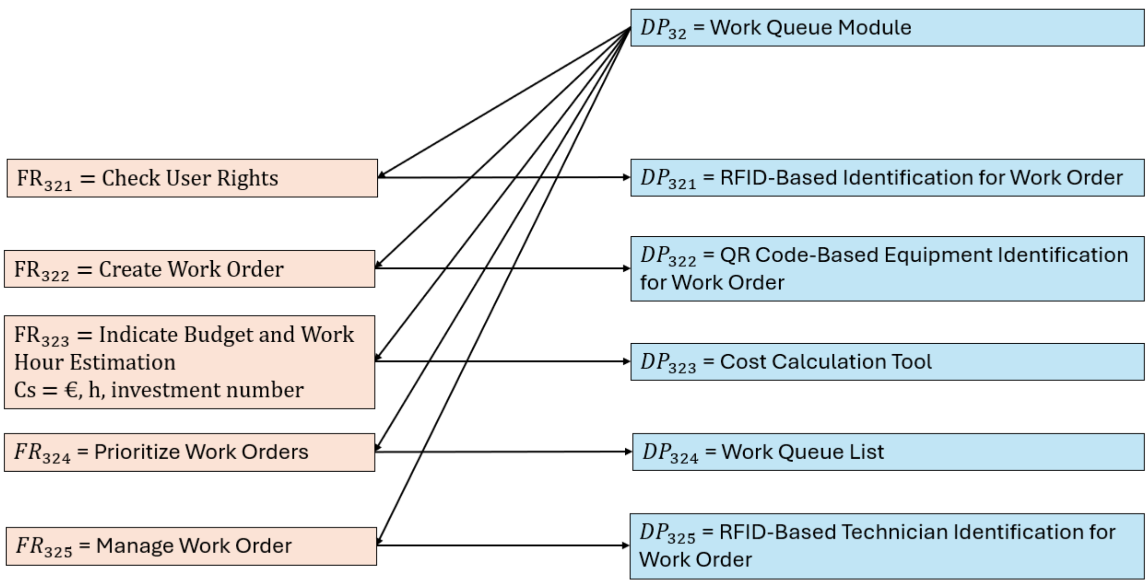


Figure 32 - DP_{32} - Work Queue Module

4.1.9 CMMS Matrix and System Architecture

The design matrix is constructed after identifying the FRs and DPs, as well as their relationships. In the design matrix, the FRs are presented on the left side as rows, while the DPs are presented across the top as columns. If a relationship exists between an FR and a DP, it is represented by a non-zero value in the corresponding matrix cell.

	DP0: Design CMMS	DP1: Application for PM Tasks	DP11: Maintenance Technician Module for PM Tasks	DP111: QR Code-Based Equipment Identification in PM Task Execution	DP112: RFID-Based Technician Identification in PM Tasks	DP113: Automatic Setup for PM Tasks	DP12: Operator Module for Daily PM Tasks	DP121: QR Code-Based Equipment Identification in Daily Inspections	DP122: RFID-Based Operator Identification in Daily Inspections	DP123: Automatic Setup for Daily Inspections	DP2: Application for Equipment Failures	DP21: Reporting Module	DP211: QR Code-Based Equipment Identification in Failure Reporting	DP212: RFID-Based Operator Identification in Failure Reporting	DP213: Automatic Information Setup to Report about a Failure	DP22: Maintenance Technician Module for Corrective Actions	DP221: Automatic Notification about a Failure from the Information Setup	DP222: QR Code-Based Equipment Identification under a Failure	DP223: RFID-Based Technician Identification for Corrective Action	DP224: Failure Management Setup	DP23: Management Module	DP231: Non-Conformance Report for Equipment	DP232: Chart of Equipment Failure Statistics Based on KPIs	DP3: Application for Scheduling and Prioritization	DP31: Maintenance Supervision Module	DP311: Automatic Prioritization of RM Tasks (FMEA)	DP312: Admin Tool for Work Queue	DP32: Work Queue Module	DP321: RFID-Based Identification for Work Order	DP322: QR Code-Based Equipment Identification for Work Order	DP323: Cost Calculation Tool	DP324: Work Queue List	DP325: RFID-Based Technician Identification for Work Order
FR0: Create CMMS	x																																
FR1: Manage PM Tasks		x																															
FR11: Perform PM Tasks			x																														
FR111: Select Correct Equipment for PM Task				x																													
FR112: Traceability for PM Tasks					x																												
FR113: Carry Out PM Tasks						x																											
FR12: Perform Daily Inspections							x																										
FR121: Select Correct Equipment for Daily Inspections								x																									
FR122: Traceability for Daily Inspections									x																								
FR123: Carry Out Daily Inspections										x																							
FR2: Manage Equipment Failures											x																						
FR21: Inform about Failures												x																					
FR211: Select Equipment under a Failure													x																				
FR212: Traceability for an Equipment Failure														x																			
FR213: Inform about a Failure															x																		
FR22: Perform Corrective Actions																x																	
FR221: Get Information about a Failure																	x																
FR222: Select Equipment to Start Corrective Action																		x															
FR223: Undertake Assignment																			x														
FR224: Diagnosis / Localization of Failure and Documentation of Corrective Actions																				x													
FR23: Follow Up Failures																					x												
FR231 = Status of PM Tasks																						x											
FR232: Access to Equipment Failure Data																						x	x										
FR3: Manage Resources																								x									
FR31: Prioritize and Schedule Tasks																									x								
FR311: Set Priority and Schedule for RM Tasks																										x	x						
FR312: Change Prioritization and Scheduling for Maintenance Tasks																											x						
FR32: Create Work and Change Orders																									x			x					
FR321: Check User Rights																													x				
FR322: Create Work Order																														x			
FR323: Indicate Budget and Work Hour Estimation																															x		
FR324: Prioritize work orders																																x	
FR325: Manage Work Order																																	x

Table 8 - Initial design matrix for CMMS.

When the same DPs are used to satisfy multiple FRs, the independence axiom is not fulfilled (see Ch. 2.2.2 – Design Matrix). This must be considered in the design, as modifications to DPs may affect multiple FRs through their interdependencies.

As shown in Table 8, DP_{312} is related to two different FRs (FR_{311} and FR_{312}), and one of the non-zero values is located above the diagonal of the design matrix. This indicates that the design is decoupled and that reordering of rows and columns is required.

To achieve a proper sequence, DP_{312} should be moved to the left, and FR_{312} should be moved one row up, placing them before the FR_{311} and DP_{311}.

	DP0: Design CMMS	DP1: Application for PM Tasks	DP11: Maintenance Technician Module for PM Tasks	DP111: QR Code-Based Equipment Identification in PM Task Execution	DP112: RFID-Based Technician Identification in PM Tasks	DP113: Automatic Setup for PM Tasks	DP12: Operator Module for Daily PM Tasks	DP121: QR Code-Based Equipment Identification in Daily Inspections	DP122: RFID-Based Operator Identification in Daily Inspections	DP123: Automatic Setup for Daily Inspections	DP2: Application for Equipment Failures	DP21: Reporting Module	DP211: QR Code-Based Equipment Identification in Failure Reporting	DP212: RFID-Based Operator Identification in Failure Reporting	DP213: Automatic Information Setup to Report about a Failure	DP22: Maintenance Technician Module for Corrective Actions	DP221: Automatic Notification about a Failure from the Information Setup	DP222: QR Code-Based Equipment Identification under a Failure	DP223: RFID-Based Technician Identification for Corrective Action	DP224: Failure Management Setup	DP23: Management Module	DP231: Non-Conformance Report for Equipment	DP232: Chart of Equipment Failure Statistics Based on KPIs	DP3: Application for Scheduling and Prioritization	DP31: Maintenance Supervision Module	DP312: Admin Tool for Work Queue	DP311: Automatic Prioritization of RM Tasks (FMEA)	DP32: Work Queue Module	DP321: RFID-Based Identification for Work Order	DP322: QR Code-Based Equipment Identification for Work Order	DP323: Cost Calculation Tool	DP324: Work Queue List	DP325: RFID-Based Technician Identification for Work Order
FR0: Create CMMS	x																																
FR1: Manage PM Tasks		x																															
FR11: Perform PM Tasks			x																														
FR111: Select Correct Equipment for PM Task				x																													
FR112: Traceability for PM Tasks					x																												
FR113: Carry Out PM Tasks						x																											
FR12: Perform Daily Inspections							x																										
FR121: Select Correct Equipment for Daily Inspections								x																									
FR122: Traceability for Daily Inspections									x																								
FR123: Carry Out Daily Inspections										x																							
FR2: Manage Equipment Failures											x																						
FR21: Inform about Failures												x																					
FR211: Select Equipment under a Failure													x																				
FR212: Traceability for an Equipment Failure														x																			
FR213: Inform about a Failure															x																		
FR22: Perform Corrective Actions																x																	
FR221: Get Information about a Failure																	x																
FR222: Select Equipment to Start Corrective Action																		x															
FR223: Undertake Assignment																			x														
FR224: Diagnosis / Localization of Failure and Documentation of Corrective Actions																				x													
FR23: Follow Up Failures																					x												
FR231 = Status of PM Tasks																						x											
FR232: Access to Equipment Failure Data																						x	x										
FR3: Manage Resources																								x									
FR31: Prioritize and Schedule Tasks																									x								
FR312: Change Prioritization and Scheduling for Maintenance Tasks																										x							
FR311: Set Priority and Schedule for RM Tasks																										x	x						
FR32: Create Work and Change Orders																									x			x					
FR321: Check User Rights																													x				
FR322: Create Work Order																														x			
FR323: Indicate Budget and Work Hour Estimation																															x		
FR324: Prioritize work orders																																x	
FR325: Manage Work Order																																	x

Table 9 - Reordered Triangular Matrix. FR_{312} and DP_{312} (highlighted in green) are sequenced before FR_{311} and DP_{311} to achieve proper design order.

As introduced earlier (Ch. 2.2.3 – System Hierarchy), the system can be structured into modules and their operational sequence by using a flow diagram. A flow diagram helps to organize the system into "*its operational sequence*" (Suh, 2001, pp. 212 - 213). In the flow diagram, the system architecture is structured starting from lower-lever modules, which act as inputs to the higher-lever modules, ultimately forming the overall system output.

Based on the design matrix presented above (Table 9), the following flow diagram for the CMMS can be constructed:

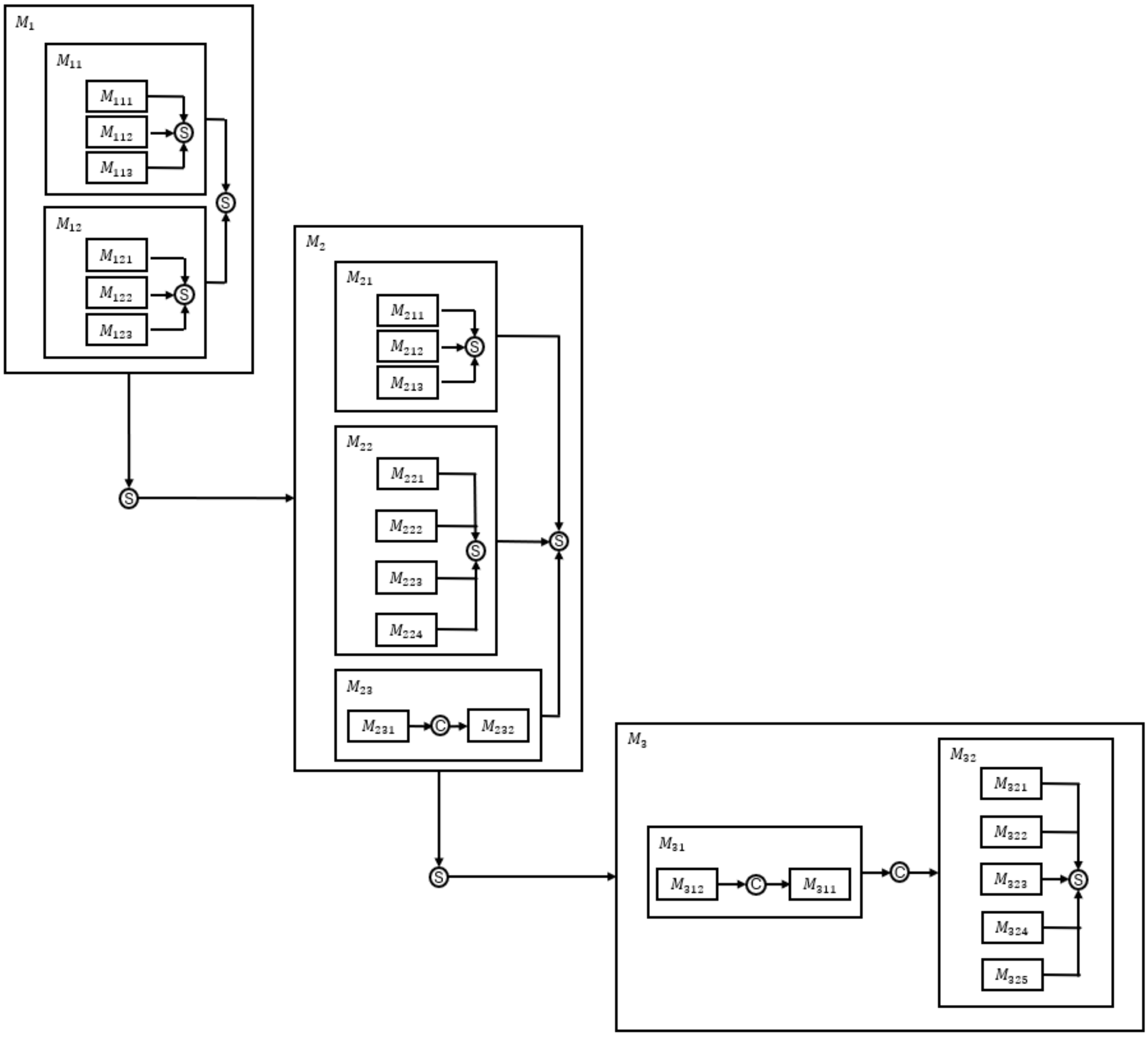


Figure 33 - Flow Diagram of the CMMS System Architecture.

As a result, all top-level modules functioning as applications within the CMMS can be operated simultaneously. For M_1, the outputs of the innermost modules (leaves) form the inputs for the higher-level modules. The presence of the junction “s” indicates that these modules can be executed in parallel.

Within M_2, the inner modules of M_{23} must be executed in series, meaning that M_{231} must be completed before M_{232}. The remaining lower-level modules within M_2 can be operated simultaneously.

In M_3, the execution sequence begins with M_{312}, followed by M_{311}, which together lead to M_{31} and subsequently to M_{32}.

5 Maintenance System Testing

To complete the study, the newly developed CMMS, created using a no-code platform, must be tested. The system should be capable of collecting relevant data on equipment usage and operation to support data-driven decision-making. As indicated in the previous chapters (Ch. 3 – Methods and Datasets and 4 – Maintenance System Development – Ideal Model), the system is designed to add value to daily operations.

This added value can be measured in terms of time and cost savings, even as the volume of available information increases and becomes more transparent to stakeholders. The testing of the CMMS should demonstrate ease of use, enable effective prioritization of maintenance tasks, and present KPIs clearly for management.

In summary, the CMMS should reduce operational waste – such as manually performed and duplicated work – by automating daily maintenance processes.

5.1 Test Plan

The CMMS will be developed according to the designed system architecture (Ch. 4.1.9 – CMMS Matrix and System Architecture). After implementation, the system will be tested by four different stakeholder groups:

1. Production operators (C2)
2. Maintenance technicians (C1)
3. Maintenance supervisor (C3)
4. Management team (C4 – C7)

The production operators (C2) focus on daily inspections and the usability of failure reporting, as well as the clarity of communication. Maintenance technicians (C1) evaluate the system by using it to perform and report PM tasks, handle equipment failures, and execute work orders created by other departments. The maintenance supervisor (C3) assesses task prioritization, work queue management, and the coordination of maintenance resources. The management team (C4 – C7) tests the system's capability to provide real-time information on asset status, monitor KPI data, and support future decision-making.

The testing will be conducted over one month to refine the system based on user feedback.

5.1.1 Test Plan for Production Operators (C2)

The test plan for production operators is divided into two sections, in which two modules used by the operators are evaluated. Both modules include test items that are verified through practical use of the system during operation.

The Daily Inspection Module (DP_{12}) should:

1. Provide a unique ID for each event. ($DP_{123} \rightarrow PV_{123}$)
2. Automatically register the date and time of the event. ($DP_{123} \rightarrow PV_{124}$)
3. Identify the relevant equipment using a QR code and prevent the selection of incorrect equipment, whether intentional or unintentional. ($DP_{121} \rightarrow PV_{121}$)
4. Register the user performing the inspection via an RFID tag and prevent the selection of another user. ($DP_{122} \rightarrow PV_{122}$)
5. List the required inspection tasks and allow the operator to indicate whether each task has been completed. ($DP_{123} \rightarrow PV_{125}$)

The Failure Reporting Module (DP_{21}) should:

1. Provide a unique ID for each event. ($DP_{213} \rightarrow PV_{213}$)
2. Automatically register the date and time of the event (required for MTBF and MTTR calculations). ($DP_{213} \rightarrow PV_{214}$)
3. Identify the relevant equipment using a QR code and prevent the selection of incorrect equipment. ($DP_{211} \rightarrow PV_{211}$)
4. Register the user reporting the failure via an RFID tag and prevent the selection of another user. ($DP_{212} \rightarrow PV_{212}$)
5. Allow the operator to describe the affected component and the nature of the failure. ($DP_{213} \rightarrow PV_{215}$)
6. Indicate the priority level of the failure and whether the equipment can continue to be used. ($DP_{213} \rightarrow PV_{216}$)

5.1.2 Test Plan for Maintenance Technicians (C1)

Maintenance technicians will use the system to report PM tasks, manage equipment failures, and execute work orders. Additionally, the system should support work planning activities.

The Maintenance Technician Module for PM Tasks (DP_{11}) should:

1. Provide a unique ID for each PM event. ($DP_{113} \rightarrow PV_{113}$)
2. Automatically register the date and time of the event. ($DP_{113} \rightarrow PV_{114}$)
3. Guide upcoming tasks based on the selected equipment and preconfigured service schedule. ($DP_{113} \rightarrow PV_{115}$)
4. Identify the relevant equipment using a QR code and prevent the selection of incorrect equipment. ($DP_{111} \rightarrow PV_{111}$)
5. Register the technician performing the inspection via an RFID tag and prevent the selection of another user. ($DP_{112} \rightarrow PV_{112}$)
6. List the PM tasks to be completed. ($DP_{113} \rightarrow PV_{116}$)
7. Allow only authorized users to complete more demanding PM tasks. ($DP_{113} \rightarrow PV_{117}$)
8. Enable the technician to add notes. ($DP_{113} \rightarrow PV_{118}$)

The Maintenance Technician Module for Corrective Actions (DP_{22}) should:

1. Send notifications about a failure. ($DP_{221} \rightarrow PV_{221}$)
2. Provide access to information on reported failures. ($DP_{224} \rightarrow PV_{224}$)
3. Display the priority of open failure tasks. ($DP_{224} \rightarrow PV_{225}$)
4. Identify the relevant equipment using a QR code and prevent the selection of incorrect equipment. ($DP_{222} \rightarrow PV_{222}$)
5. Register the technician via an RFID tag and prevent the selection of another user. ($DP_{223} \rightarrow PV_{223}$)

6. Enable the start of repair work and record reaction time and repair time (as required for MTTR calculations). ($DP_{224} \rightarrow PV_{226}$)

7. Allow the technician to set the status to "on hold" when spare parts are ordered. ($DP_{224} \rightarrow PV_{227}$)

8. Allow the technician to continue the repair work. ($DP_{224} \rightarrow PV_{228}$)

9. Allow the technician to complete the task and document corrective actions. ($DP_{224} \rightarrow PV_{229}$)

5.1.3 Test Plan for Maintenance Supervisor (C3)

The Maintenance Supervision Module (DP_{31}) should:

1. Allow the supervisor to reprioritize tasks. ($DP_{31} \rightarrow PV_{31}$)

5.1.4 Test Plan for Maintenance Department (C1; C3)

The Work Queue Module (DP_{32}) should:

1. Identify the relevant equipment using QR code and prevent the selection of incorrect equipment. ($DP_{322} \rightarrow PV_{322}$)

2. Register the user performing the task via RFID tag and prevent the selection of another user. ($DP_{325} \rightarrow PV_{325}$)

3. Allow only authorized users to fulfill work orders. ($DP_{321} \rightarrow PV_{321}$)

4. Enable the user to enter project number, budget and estimated working hours into the work order form. ($DP_{323} \rightarrow PV_{323}$)

5. Create a list of maintenance tasks and provide management with visibility of their status. ($DP_{324} \rightarrow PV_{324}$)

6. Allow the maintenance supervisor to reprioritize the task list. ($DP_{312} \rightarrow PV_{312}$)

7. Allow the maintenance supervisor to reject or approve proposed work orders. ($DP_{324} \rightarrow PV_{326}$)

8. Enable the maintenance supervisor to assign high or low prioritization to work orders. $(DP_{324} \rightarrow PV_{327})$
9. Allow the maintenance technician to select and undertake a task from the list according to prioritization. $(DP_{324} \rightarrow PV_{328})$
10. Measure the time spent completing the work order. $(DP_{324} \rightarrow PV_{329})$
11. Allow the technician to set the status to "on hold" when spare parts are ordered. $(DP_{324} \rightarrow PV_{330})$
12. Allow the technician to continue the work order. $(DP_{324} \rightarrow PV_{331})$
13. Allow the technician to set the status to "complete" and document the expenses for components, as well as add notes according to the work order. $(DP_{32} \rightarrow PV_{332})$

5.1.5 Test Plan for Management Team (C4 – C7)

The management team should be able to obtain the following information using the Management Module (DP_{23}):

1. Daily inspection reports of equipment with "NOK" (not okay) statuses. $(DP_{231} \rightarrow PV_{231})$
2. Equipment failure statistics, including relevant KPIs for monitoring performance. $DP_{232} \rightarrow PV_{232})$

5.2 Test Evaluation Criteria

The test results can be evaluated based on the performance and successful completion of the test items outlined in the test plan. Additionally, user feedback can be used to assess the system according to the following assumptions:

1. Ease of Use: If data is gathered weekly during the testing period, the system's usability can be assessed by how quickly users can generate reports and navigate the applications.
2. Management Access: Management should be able to locate information about the equipment responsible for the most failures in the past week in under one minute.

3. Data visualization: Charts and visualization should be clear and easy to interpret.
4. Feature Functionality: All module features defined in the test plan should function as intended.

6 Results

The pilot CMMS application was developed using the no-code platform AppSheet. The following sections describe the system design within the platform, the implemented functionalities, and the capabilities offered by the application. The system was built based on the previously defined CMMS architecture, equipment data, and test plan to ensure it meets operational requirements.

After presenting the system design and functionality, the results of the CMMS testing are discussed in Chapter 6.2 – Test Results of the CMMS.

6.1 Pilot CMMS Design with AppSheet

6.1.1 Equipment List

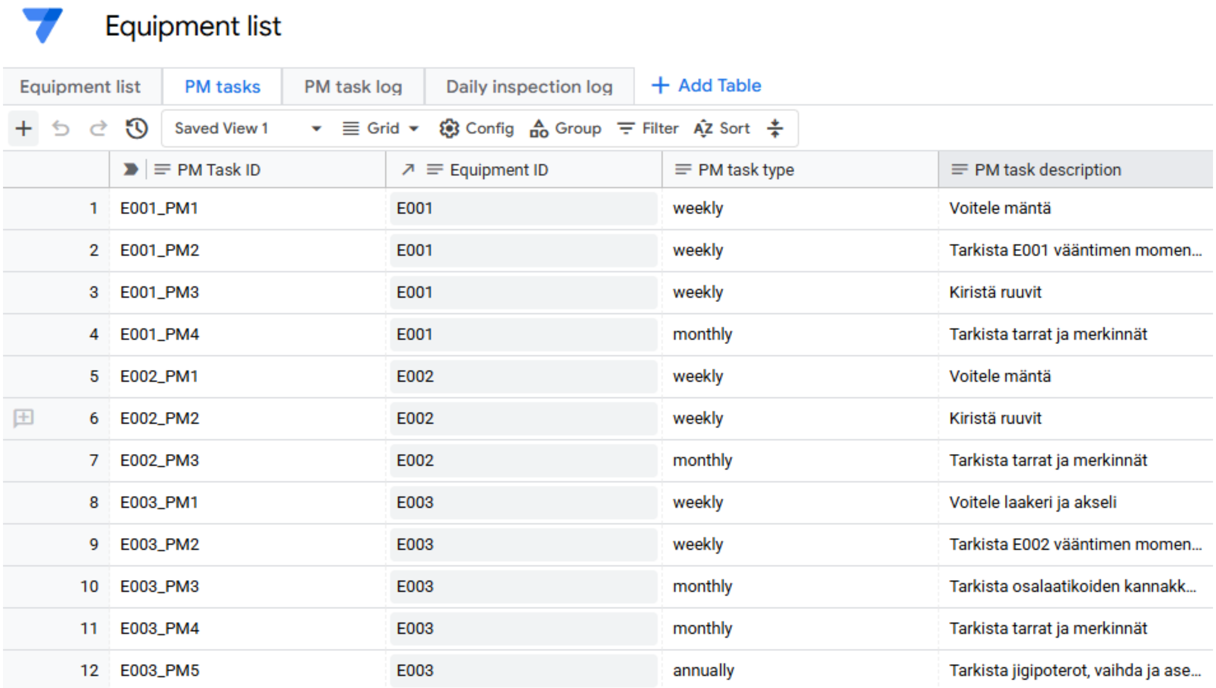
The creation of the CMMS application begins with establishing the database. A new database, called “Equipment list”, is created, containing a table that serves as a master list for all tables requiring equipment information. The table columns function as attributes for the application. At a minimum, the equipment list includes the ID number and description of each piece of equipment. Additionally, the list gathers relevant information such as the commissioning date, manufacturer, type number, and other required information that were described earlier (Ch. 3.1.6 – Maintenance Documentation and 3.2.4 – Analysis of Maintenance Documentation).

6.1.2 Maintenance Technician Module for PM Tasks (DP_{11})

DP_{11} - Maintenance Technician Module for PM Tasks is the first functionality implemented in the CMMS. A new table for PM tasks is created within the “Equipment list” database, where each row corresponds to a task associated with a specific piece of equipment.

Each task is assigned a unique “PM Task ID”, and its relationship to the equipment is established using the “Equipment ID” with the “ref” type. This configuration allows tasks to be filtered by equipment in the PM task form, ensuring that only the tasks corresponding to the selected equipment are displayed.

The table also includes a PM task type column to specify whether the task should be performed weekly, monthly, semi-annually or annually, along with a column for the task description.



Equipment list

Equipment list | PM tasks | PM task log | Daily inspection log | + Add Table

	PM Task ID	Equipment ID	PM task type	PM task description
1	E001_PM1	E001	weekly	Voitele mäntä
2	E001_PM2	E001	weekly	Tarkista E001 vääntimen momen...
3	E001_PM3	E001	weekly	Kiristä ruuvit
4	E001_PM4	E001	monthly	Tarkista tarrat ja merkinnät
5	E002_PM1	E002	weekly	Voitele mäntä
6	E002_PM2	E002	weekly	Kiristä ruuvit
7	E002_PM3	E002	monthly	Tarkista tarrat ja merkinnät
8	E003_PM1	E003	weekly	Voitele laakeri ja akseli
9	E003_PM2	E003	weekly	Tarkista E002 vääntimen momen...
10	E003_PM3	E003	monthly	Tarkista osalaatikoiden kannakk...
11	E003_PM4	E003	monthly	Tarkista tarrat ja merkinnät
12	E003_PM5	E003	annually	Tarkista jigipoterot, vaihda ja ase...

Figure 34 - PM tasks table showing each PM task associated with its respective equipment.

Technicians can begin a PM task by selecting the ‘PM Tasks’ view in the CMMS application and pressing the button with the oil can icon.

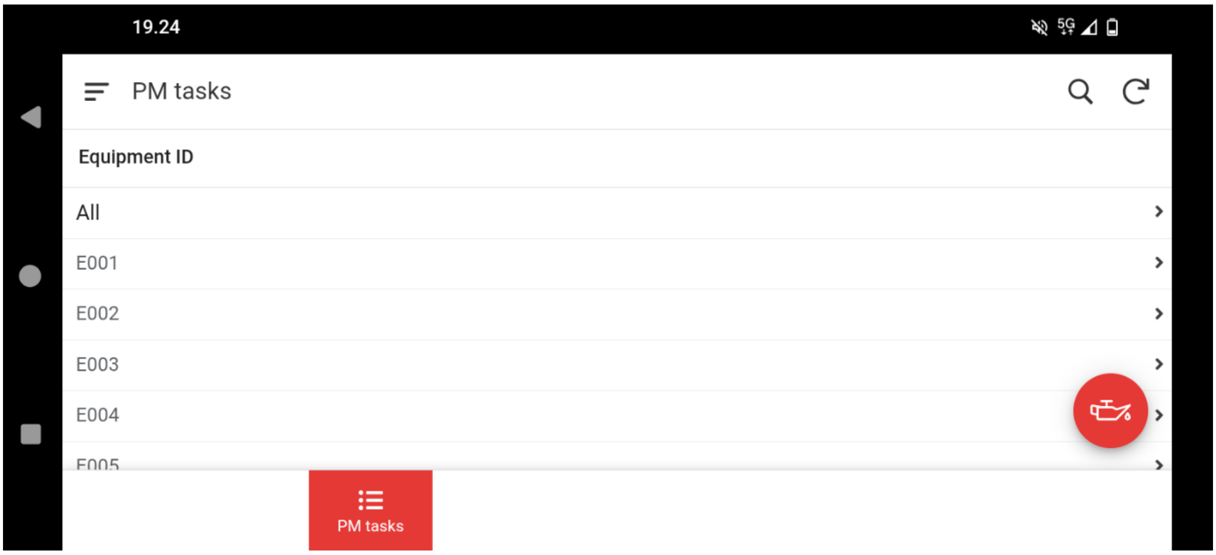


Figure 35 – ‘PM tasks’ view in the CMMS. Selecting a piece of equipment displays its PM task log. Pressing the oil can button on the lower-right corner allows the technician to start performing the PM task.

When starting a PM task, only a few options are initially displayed. After completing the required fields, the application guides the user through the next steps. The technician must scan the QR code to select the correct equipment.

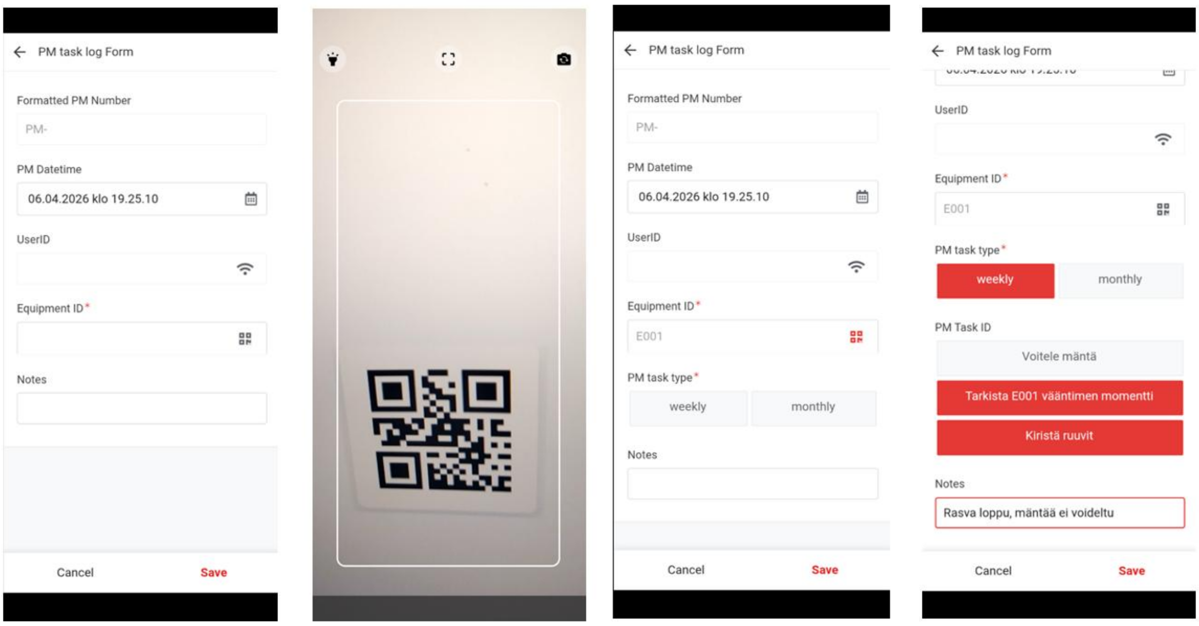


Figure 36 - PM task form. After completing and saving the form, a new row with a unique "Formatted PM number" is added to the PM task log table.

In the case of a PM event for the “E001” equipment, the application allows the user to choose between two different task types: weekly or monthly PM tasks. By selecting the weekly option, the app displays three tasks to be checked and completed. In the example above (Figure 36), the lubrication task is left undone due to lack of grease.

The completed form is then transferred via an AppSheet action to the failure log table. A new ID for the PM event, called the “Formatted PM number”, is automatically generated using the AppSheet automation feature when the form is saved.

Equipment list

Equipment list | Daily inspection log | PM tasks | PM task log | + Add Table

	PM...	Formatted PM Number	Equipment ID	PM Datetime	PM Task ID	Notes	PM task type
1	10	PM-10	E001	6.4.2026 klo 19.25	Tarkista E001 vääntimen momentti, Kiristä ruuvit	Rasva loppu, mäntää ei voideltu	weekly
2	9	PM-9	E001	21.3.2026 klo 15.44	Voitele mäntä, Tarkista E001 vääntimen momentti, Kiristä ruuvit	Tilattu lisää voiteluainetta	weekly
3	8	PM-8	E007	21.3.2026 klo 14.57	Tarkista rullat, Kiristä ruuvit		weekly
4	7	PM-7	E001	21.3.2026 klo 14.57	Voitele mäntä, Tarkista E001 vääntimen momentti, Kiristä ruuvit		weekly
5	6	PM-6	E006	21.3.2026 klo 14.56	Tarkista tarrat ja merkinnät		monthly
6	5	PM-5	E005	21.3.2026 klo 14.55	Tarkista E005 vääntimen momentti, Tarkista rullat, Voitele painimen j		weekly
7	4	PM-4	E004	21.3.2026 klo 14.55	Puhdista IV-putket ja vaihda suodatin.		annually
8	3	PM-3	E002	21.3.2026 klo 14.50	Voitele mäntä, Kiristä ruuvit		weekly
9	2	PM-2	E001	21.3.2026 klo 14.50	Voitele mäntä, Tarkista E001 vääntimen momentti, Kiristä ruuvit		weekly

Figure 37 – ‘PM task log’ -table showing the history for preventive maintenance actions.

6.1.3 Daily Inspections Module (DP_{12})

The second module to be implemented is the DP_{12} – Daily Inspection Module. In this module, the operator can create a new inspection record by filling out a form. The daily inspection is initiated by pressing the “Check” button.

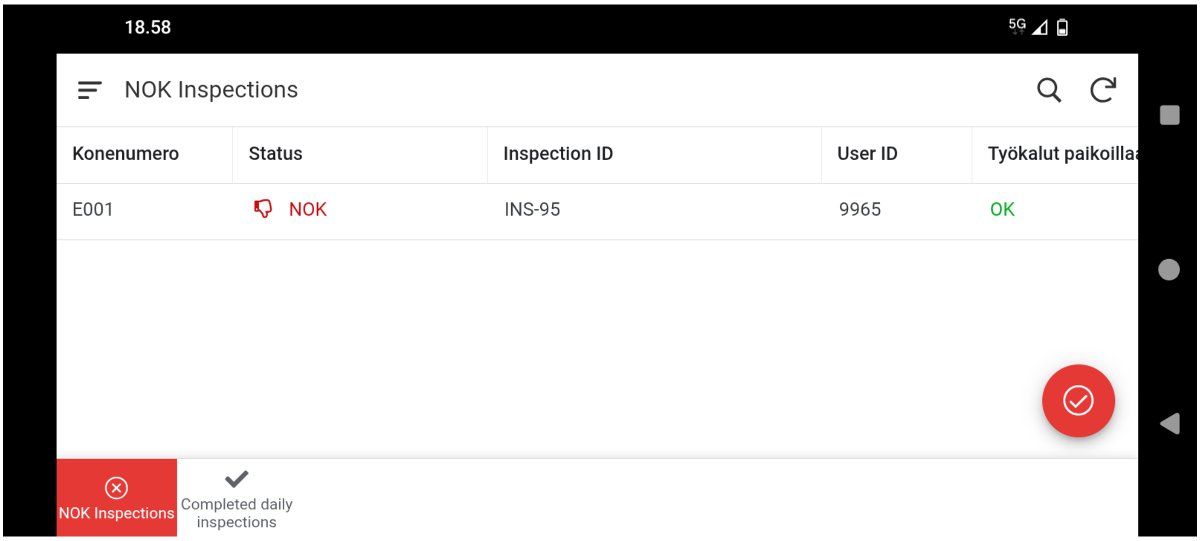


Figure 38 - The daily inspection can be started by pressing the “Check” button on the lower-right corner. If the inspection cannot be fully completed, it will appear in the "NOK Inspection” view as shown.

The inspection form collects information such as the inspection ID, date and time, answers to inspection questions, and any notes provided by the operator.

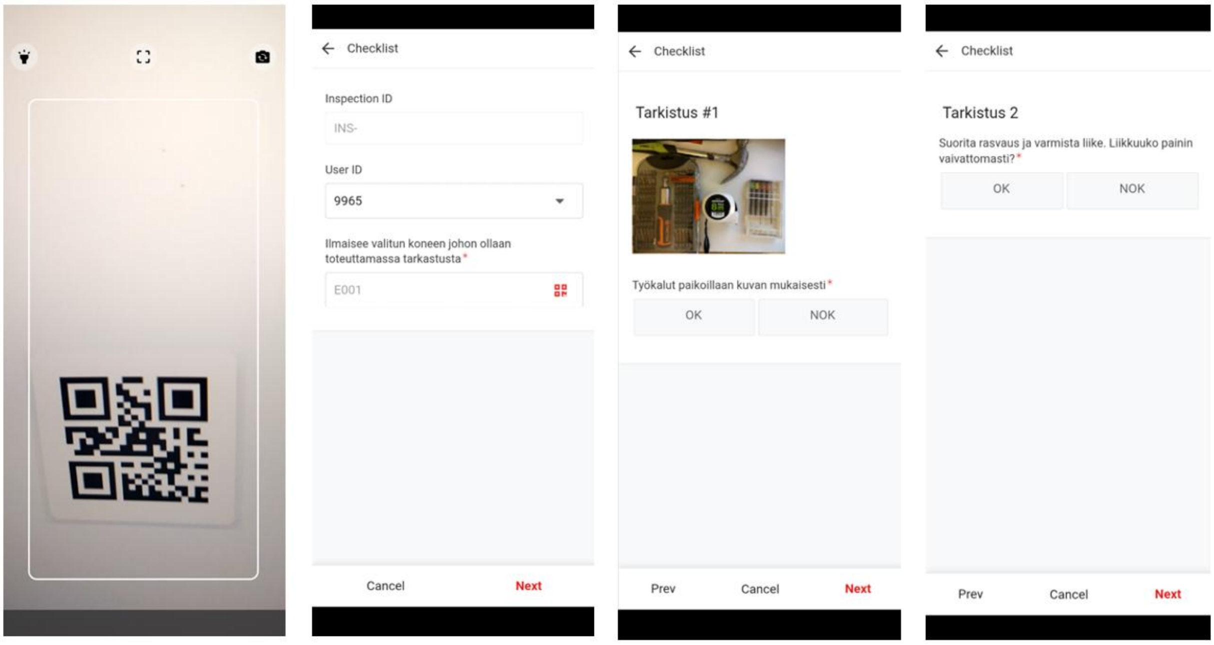


Figure 39 - Daily inspection form, requiring all fields to be completed before proceeding to the next page.

If any inspection item does not receive an “OK” status, the inspection is not considered complete and will be listed in the “NOK inspections” view. This allows management to monitor nonconforming conditions in equipment.

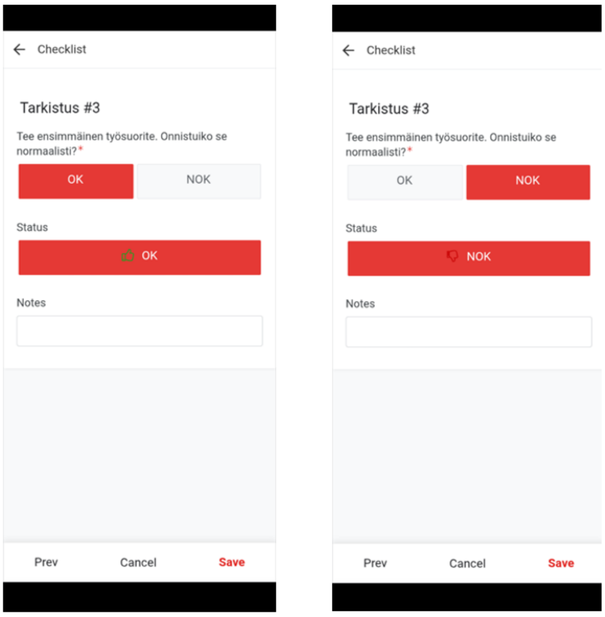
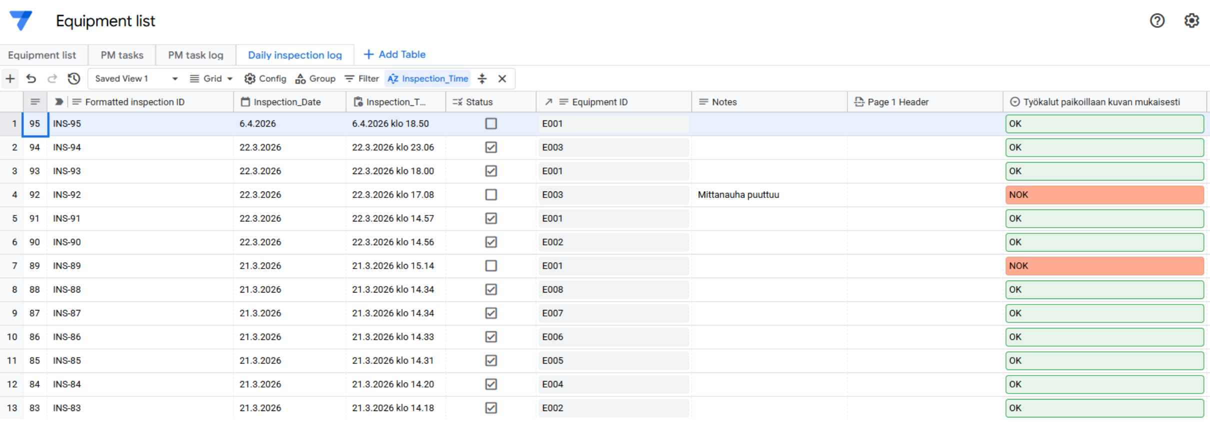


Figure 40 - Status overview of the daily inspection form, displayed on the final page before saving and submitting the results.

Once the inspection is completed, the results are saved as a new row in the 'Daily inspection log' table.



		Formatted inspection ID	Inspection_Date	Inspection_T...	Status	Equipment ID	Notes	Page 1 Header	Työkalut paikoillaan kuvan mukaisesti
1	95	INS-95	6.4.2026	6.4.2026 klo 18.50	☐	E001			OK
2	94	INS-94	22.3.2026	22.3.2026 klo 23.06	☑	E003			OK
3	93	INS-93	22.3.2026	22.3.2026 klo 18.00	☑	E001			OK
4	92	INS-92	22.3.2026	22.3.2026 klo 17.08	☐	E003	Mittanauha puuttuu		NOK
5	91	INS-91	22.3.2026	22.3.2026 klo 14.57	☑	E001			OK
6	90	INS-90	22.3.2026	22.3.2026 klo 14.56	☑	E002			OK
7	89	INS-89	21.3.2026	21.3.2026 klo 15.14	☐	E001			NOK
8	88	INS-88	21.3.2026	21.3.2026 klo 14.34	☑	E008			OK
9	87	INS-87	21.3.2026	21.3.2026 klo 14.34	☑	E007			OK
10	86	INS-86	21.3.2026	21.3.2026 klo 14.33	☑	E006			OK
11	85	INS-85	21.3.2026	21.3.2026 klo 14.31	☑	E005			OK
12	84	INS-84	21.3.2026	21.3.2026 klo 14.20	☑	E004			OK
13	83	INS-83	21.3.2026	21.3.2026 klo 14.18	☑	E002			OK

Figure 41 – Completed daily inspection results saved in the 'daily inspection log' table.

6.1.4 Equipment Failure Reporting Module (DP_{21})

A new database is created to manage equipment failures. The equipment failure reporting process is divided into several phases:

1. Reporting phase
2. Repair phase
3. Spare part ordering phase (if needed)
4. Completion phase

New failures can be reported from the ‘New Failures’ view. By pressing the button with an ‘exclamation triangle’ icon, operators can submit a new failure report.

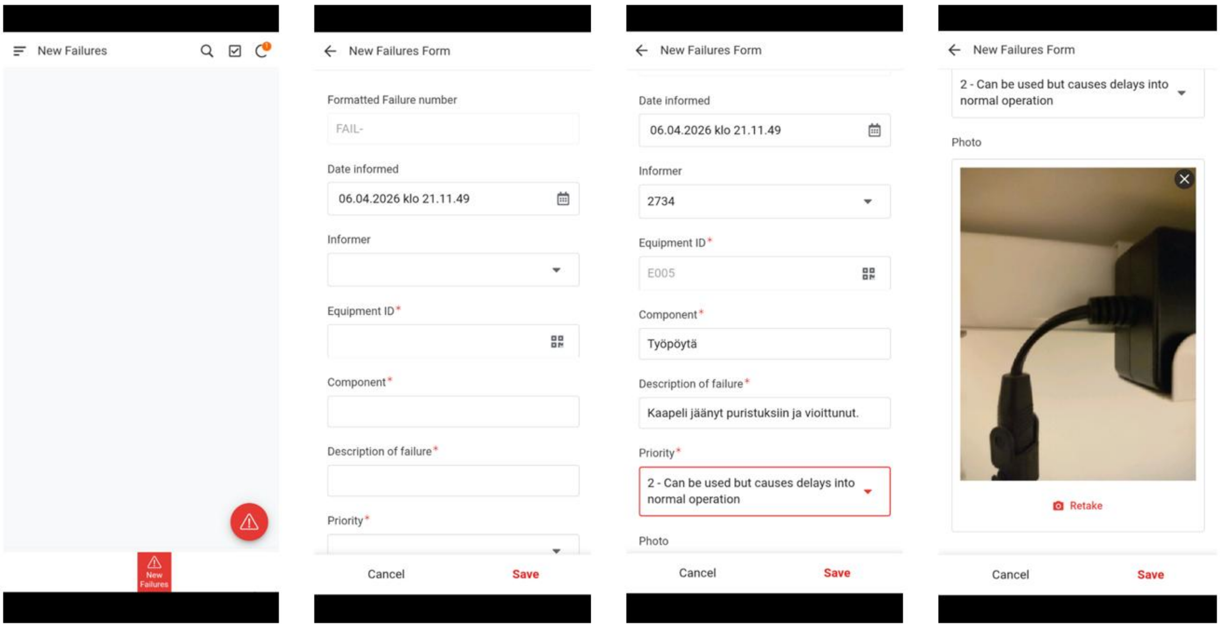


Figure 42 – The failure report form requires all mandatory fields to be completed before submission. A picture can also be attached to provide more detailed information about failure location.

Once submitted, the new failure reports are assigned a “Reported” status and listed in the ‘New Failures’ view.

6.1.5 Equipment Failure Corrective Actions Module (DP_{22})

Reports are arranged by priority, helping technicians determine which tasks to address first. By selecting a report from the ‘New Failures’ view, the technician can view detailed information about the failure and initiate the repair action directly from this detailed view by pressing the “Repair” button.

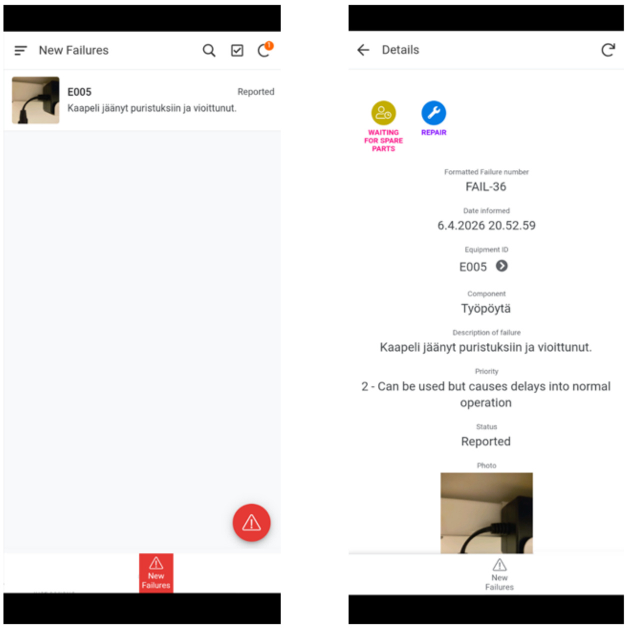


Figure 43 - The technician can initiate corrective actions for a reported failure from the detail view by pressing the blue "Repair" button.

The repair phase requires the user to have a technician role. Pressing the “Repair” button initiates the corrective action and moves the report to the “Under Repair” view, where all active reports are listed. Tasks in this view are either actively being repaired or are marked as “Waiting for Spare Parts”. When a task is on hold for spare parts, repair time is not recorded. Once the spare parts arrive, the technician presses the “Continue” button to resume the repair process, ensuring that repair time is only measured when work is actively performed.

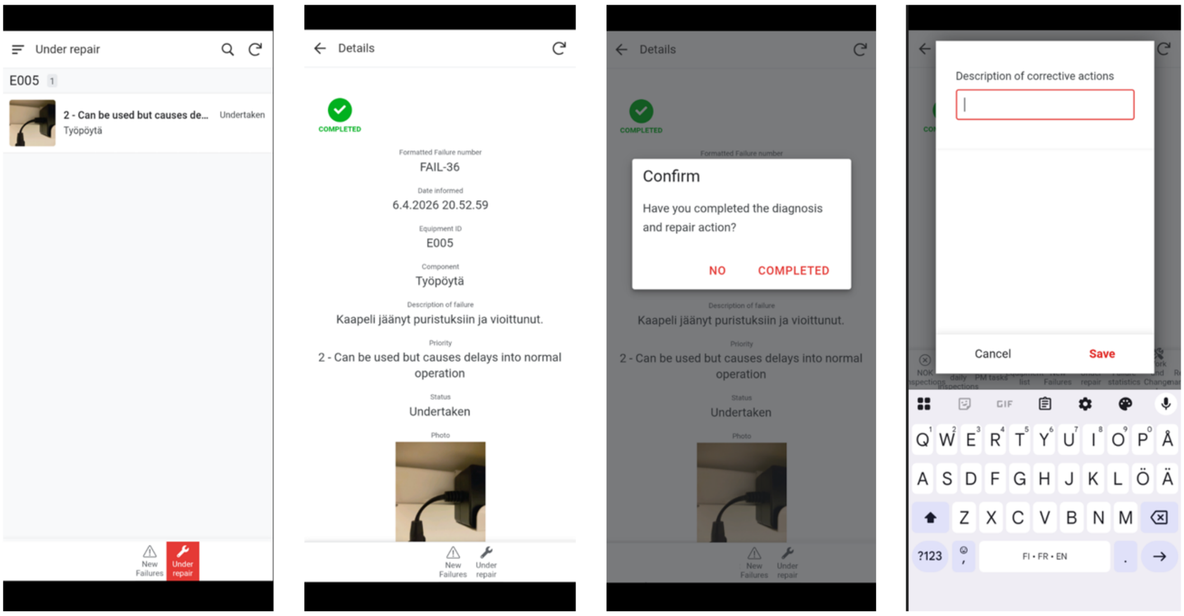


Figure 44 - Corrective actions for a reported failure can be completed from the incident’s detail view.

When the repair process is completed, the technician stops the repair time recording from the incident's detail view by pressing "Completed" button. The system then prompts the user to provide their identification and a description of the corrective action.

6.1.6 Management Module (DP_{23})

The management team can assess failure data through the KPI dashboard. This dashboard presents charts for key equipment metrics such as MTBF, MTTR, and availability, as introduced earlier (Ch. 2.1.6 – Mean Time Between Failures, MTBF; 0 – Equation 1 – MTBF.

Mean Time to Repair, MTTR; 2.1.8 – Overall Equipment Effectiveness, OEE). While detailed equipment-specific figures can be viewed in the equipment list, the KPI dashboard provides a clear graphical overview of the current status.

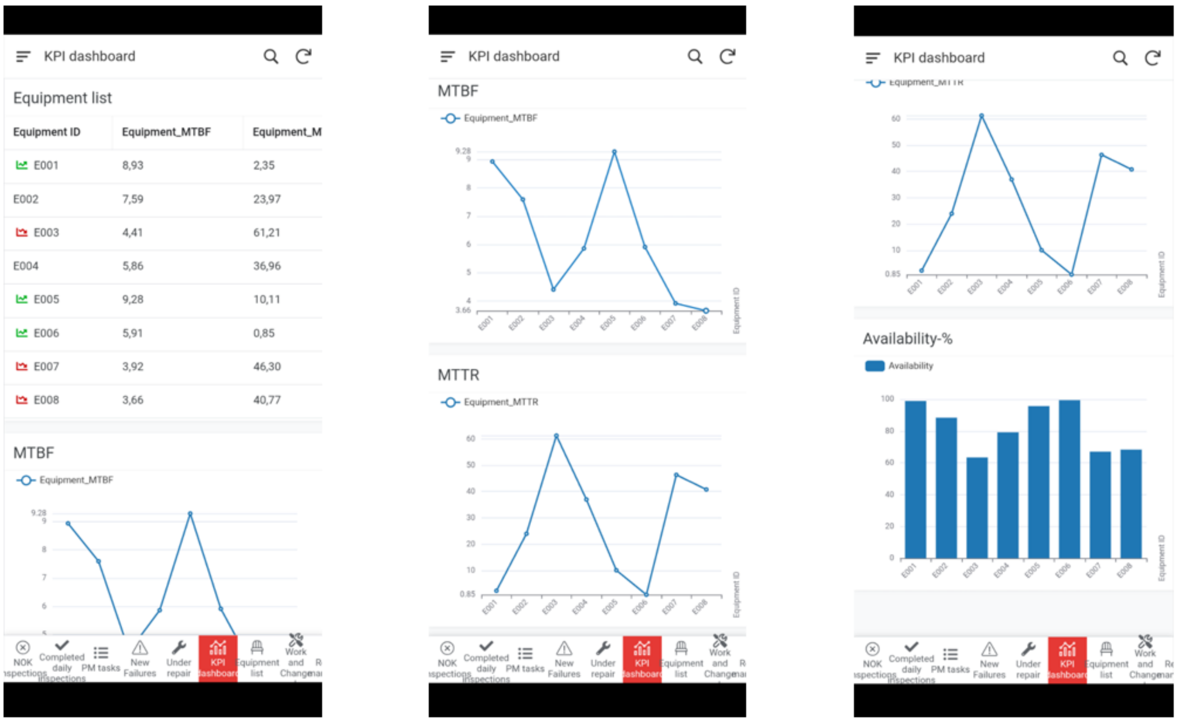


Figure 45 - The KPI dashboard displaying equipment failure rates and efficiency, providing live status for management.

AppSheet's formatting rules can be used to highlight important measures with icons and colors. As shown in Figure 45, the equipment list uses green and red indicators to represent availability levels. For example, if availability is below 75%, the indicator before the "Equipment ID" appears red; and if availability is above 95%, it appears green.

6.1.7 Maintenance Supervisor Module (DP_{31}) and Work Queue Module (DP_{32})

The management team can create work and change orders using the CMMS. Each work order has a dedicated form, accessible through the ‘Work and Change Orders’ view. The work order process is divided into six stages:

1. Proposal for work order.
2. Approval and prioritization of the work order.
3. Execution of the work order.
4. Spare part ordering (if required).
5. Continuation of the work order.
6. Completion of the work order.

In the first stage, the proposer must:

- Select the correct equipment for the requested change (QR code scan).
- Log in as an authenticated user to submit the work order.
- Provide a project number and work order description.
- Estimate the effort required by specifying working hours and budget.
- Specify a target completion date.

log list | Work and change orders | Resource Management | Copy of Failure log list | + Add Table

Saved View 1 | Grid | Config | Group | Filter | Sort

	Order ID	Page 1 Header	Formatted order num...	Order date	Project number	Order description	Proposed comp...	Budget
1	1		ORD-1	29.3.2026 klo 13.09	Continuous Improvement	Asennettava lisävalaisin	29.3.2026	50,00 $
2	2		ORD-2	29.3.2026 klo 13.13	Continuous improvement	Tarvitaan uusi työkalu mutterin puristamiseen	29.3.2026	150,00 $
3	3		ORD-3	30.3.2026 klo 17.16	Continuous improvement	Lisätään jarru hidastamaan pyöritystä	30.3.2026	100,00 $
4	4		ORD-4	30.3.2026 klo 17.22	Testi1	Testi1	30.3.2026	
5	5		ORD-5	30.3.2026 klo 18.01	Testi2	Testi2	30.3.2026	
6	6		ORD-6	30.3.2026 klo 18.35	Testi3	Testi3	30.3.2026	50,00 $
7	7		ORD-7	3.4.2026 klo 17.38	Continuous improvement	Lattiamerkinnät pakkausasemalle	6.4.2026	75,00 $
8	8		ORD-8	3.4.2026 klo 17.42	Continuous improvement	Lattiamerkinnät E001 esikokoonpanolle	3.4.2026	
9	9		ORD-9	3.4.2026 klo 17.48	Continuous improvement	Pöytämerkinnät E001 esikokoonpanoon	3.4.2026	40,00 $
10	10		ORD-10	3.4.2026 klo 19.00	Testausta	Lopputestausta	3.4.2026	
11	11		ORD-11	3.4.2026 klo 19.19	Työtilaustesti	Työtilaustesti koontariville1	3.4.2026	60,00 $

Figure 46 - 'Work Order' table displaying the created orders.

In the proposal stage, the user selects the correct equipment (QR code scan), logs in as an authenticated user, provides a project number and description, estimates working hours and budget, sets a completion date, and can attach supporting files. Once complete, the work order is submitted for maintenance supervisor review.

The maintenance supervisor can approve or reject the proposal and assign a priority, which determines the work queue for technicians. Approved orders proceed to execution, where technicians perform tasks and the system records working hours.

If spare parts or other components are needed, the work order is placed on "Waiting for Spare Parts" status, during which working hours are not tracked. Once parts arrive, the technician resumes work and updates the status to "Continuing Work".

Upon completion, the technician marks the work order as complete and documents any changes, costs, or notes. The CMMS automation then calculates the total hours spent on the task.

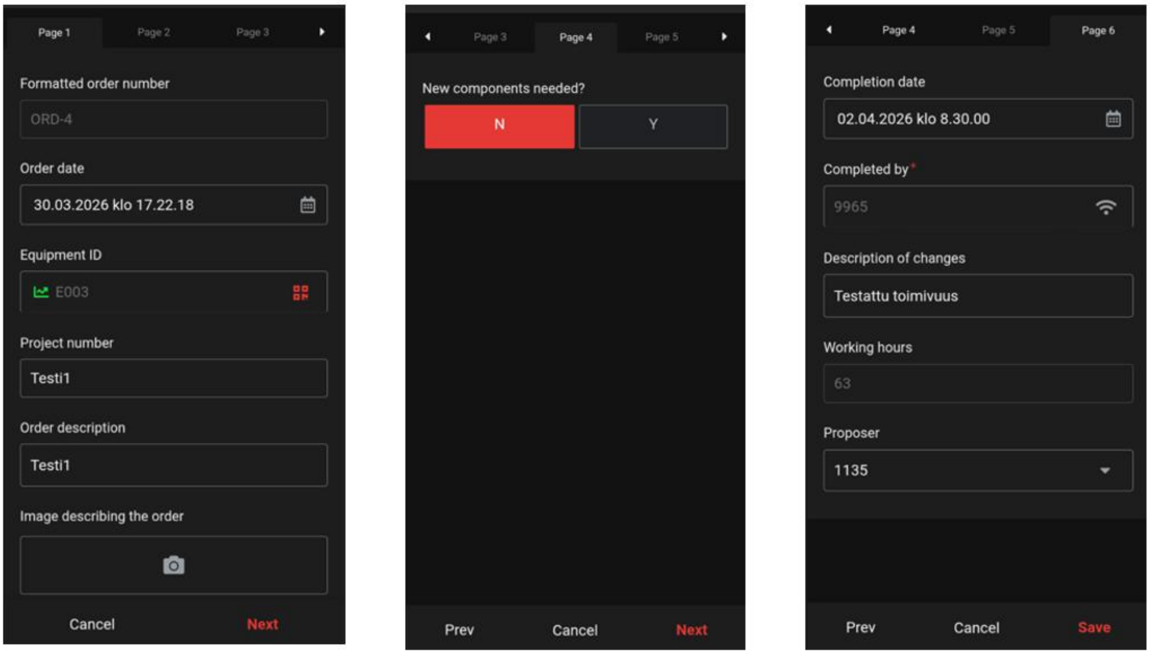


Figure 47 – Completion of a work order in the CMMS application.

6.1.8 User Roles

User access within the CMMS is managed through a centralized user list, where each individual is assigned a specific role. Four distinct user classes have been established: Admin, Maintenance Technician, Operator, and Manager.

Maintenance Technicians (C1) are authorized to execute PM tasks, including both those assigned to technicians and those allocated to operators, such as daily inspections. They are also responsible for reporting equipment failures, performing corrective actions, and completing assigned work orders.

Operators (C2) have visibility of all system views, but are restricted to conducting daily inspections and submitting failure reports, consistent with their operational role.

Maintenance Supervisor (C3), designated as system administrator, possesses full access to all CMMS functionalities. The supervisor’s primary responsibility is the approval and prioritization of the work orders, which are subsequently assigned to appropriate personnel.

Managers (C4 – C7) have access to propose work orders and monitor equipment performance through various dashboards and reports, enabling informed decision-making regarding operational and maintenance activities.

This role-based access structure ensures that users are granted permissions aligned with their operational responsibilities, supporting both accountability and efficient workflow management within the maintenance environment.

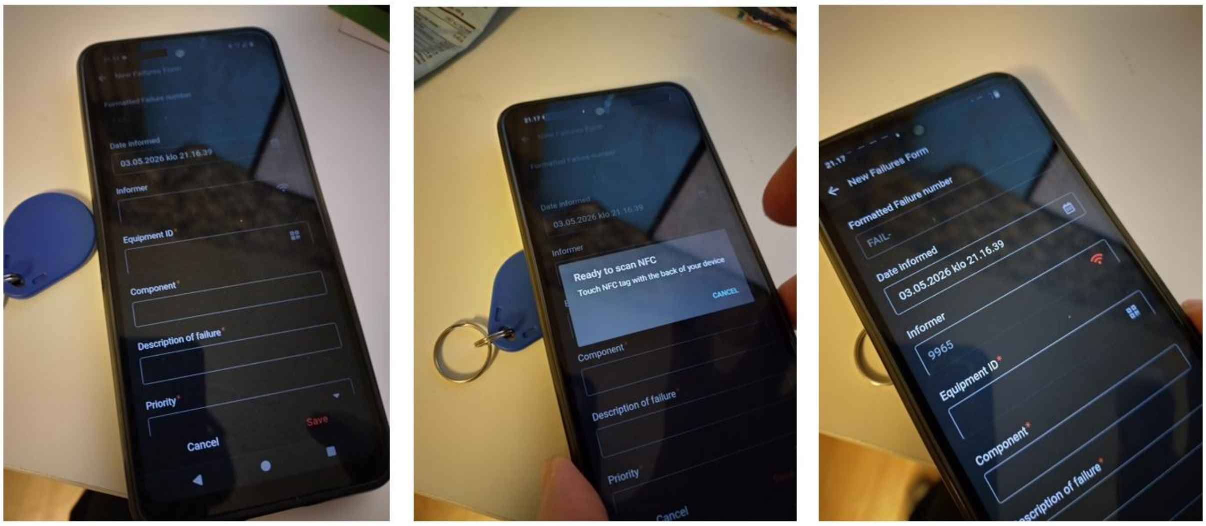


Figure 48 - DP_{212} - RFID-Based Operator Identification in Failure Reporting. The operator is identified using their personal NFC tag.

6.2 Test Results of the CMMS

Four different stakeholder groups were defined for the CMMS testing in the test plan: Production Operators (C2), Maintenance Technicians (C1), Maintenance Supervisor (C3), and the Management Team (C4 – C7).

6.2.1 Test Results of Production Operators (C2)

The testing of Production Operators (C2) was performed by the author. Two modules were tested: DP_{12} and DP_{21}.

DP_{12} – *Operator Module for Daily PM Tasks* had five test items, all of which were successfully accomplished (5/5). The documented inspection records can be found in the 'daily inspection log' table (Figure 41). The module functionality includes:

1. Providing a unique ID for each inspection event (PV_{123}).
2. Automatically registering the date and time of each inspection event (PV_{124}).
3. Identifying and registering the relevant equipment using a specific QR code (PV_{121}).
4. Identifying and registering the event performer via NFC tag (PV_{122}).
5. Listing the required inspection tasks and allowing the operator to indicate which inspection items have been completed (PV_{125}).

DP_{21} – *Failure Reporting Module* had six test items, all of which were successfully accomplished (6/6). Figure 42 shows the failure reporting form, which also allows an optional photo attachment to describe the failure. The module functionality includes:

1. Providing a unique ID for each failure event (PV_{213}).
2. Automatically registering the date and time of the failure event (PV_{214}).
3. Identifying and registering the relevant equipment for the failure using a specific QR code (PV_{211}).
4. Identifying and registering the failure reporter via NFC tag (PV_{212}).
5. Allowing the operator to describe the affected component and provide a detailed failure description (PV_{215}).
6. Indicating the priority level of the failure and specifying whether the equipment can continue to be used (PV_{216}).

6.2.2 Test Results of Maintenance Technicians (C1)

The testing of Maintenance Technicians (C1) was performed by the author. Two modules were tested: DP_{11} and DP_{22}.

DP_{11} – *Maintenance Technician Module for PM Tasks* had eight test items, seven of which were successfully completed (7/8). The PV_{115} scheduling feature was not tested, as the service program was not fully configured. The results are documented in Figure 34 Figure 37. The module functionality includes:

1. Providing a unique ID for each PM event. (PV_{113}).
2. Automatically registering the date and time of the PM event. (PV_{114}).
3. Guiding upcoming tasks based on the selected equipment and preconfigured service schedule. (PV_{115}; not active in the current system)
4. Identifying the relevant equipment for the PM using a specific QR code and preventing the selection of incorrect equipment. (PV_{111})
5. Registering the technician performing the PM task via NFC tag and preventing the selection of another user. (PV_{112})
6. Listing the PM tasks to be completed. (PV_{116})
7. Allowing only authorized users to complete more demanding PM tasks. (PV_{117})
8. Enabling the technician to add notes. (PV_{118})

DP_{22} – *Maintenance Technician Module for Corrective Actions* had nine test items, seven of which were successfully completed (7/9). The PV_{221} feature for sending notifications about a failure was not yet implemented in the system and therefore was not tested. However, AppSheet provides an automation bot, which can be used to automatically notify users when event occurs. Additionally, PV_{222} is not used in this module, since the repair process begins from the 'New Failures' view by selecting the relevant incident.

The module functionality includes:

1. Sending notifications about a failure. (PV_{221}; not active in the current system)

2. Providing access to information on reported failures. (PV_{224})

3. Displaying the priority of open failure tasks. (PV_{225})

4. Identifying the relevant equipment for the failure using a specific QR code and preventing the selection of incorrect equipment. (PV_{222}; not active in the current system)

5. Registering the technician starting the repair via NFC tag and preventing the selection of another user. (PV_{223})

6. Enabling the start of repair work and recording reaction time and repair time (as required for MTTR calculations). (PV_{226})

7. Allowing the technician to set the status to "on hold" when spare parts are ordered. (PV_{227})

8. Allowing the technician to continue the repair work. (PV_{228})

9. Allowing the technician to complete the task and document corrective actions. (PV_{229})

6.2.3 Test Results of Maintenance Supervisor (C3)

The Maintenance Supervisor used the DP_{31} – *Maintenance Supervision Module* successfully. The module had a single test item to verify its core functionality:

1. Enabling the supervisor to reprioritize tasks. (PV_{31})

6.2.4 Test Results of Maintenance Department (C1; C3)

The author simulated and tested the DP_{32} – *Work Queue Module* (Figure 46 Figure 47). The module had 13 test items, which were all successfully completed. The module functionality includes:

1. Identifying the relevant equipment using a QR code and preventing the selection of incorrect equipment. (PV_{322})

2. Registering the user performing the task via NFC tag and preventing the selection of another user. (PV_{325})

3. Allowing only authorized users to fulfill work orders. (PV_{321})

4. Enabling the user to enter project number, budget and estimated working hours into the work order form. (PV_{323})
5. Creating a list of maintenance tasks and providing management with visibility of their status. (PV_{324})
6. Allowing the maintenance supervisor to reprioritize the task list. (PV_{312})
7. Allowing the maintenance supervisor to reject or approve proposed work orders. (PV_{326})
8. Enabling the maintenance supervisor to assign high or low prioritization to work orders. (PV_{327})
9. Allowing the maintenance technician to select and undertake a task from the list according to prioritization. (PV_{328})
10. Measuring the time spent completing the work order. (PV_{329})
11. Allowing the technician to set the status to “On Hold” when spare parts are ordered. (PV_{330})
12. Allowing the technician to continue the work order. (PV_{331})
13. Allowing the technician to set the status to “Complete” and document the expenses for components, as well as add notes according to the work order. (PV_{332})

6.2.5 Test Results of Management Team (C4 – C7)

DP_{23} – *Management Module* had two test items. Both involve system-generated data and were successfully completed (Figure 38Figure 45). The module functionality includes:

1. Displaying “NOK” statuses of daily inspection reports. (PV_{231})
2. Monitoring KPIs based on equipment failure statistics (PV_{232})

6.2.6 Evaluation of Pilot CMMS Performance

The results indicate that the created pilot CMMS fulfills the preplanned activities as designed. According to the criteria defined earlier (Ch. 5.2 – Test Evaluation Criteria) four assumptions were set to be assessed:

1. Ease of use: Consistent data flows indicate high usability.

2. Management access: Since the views displaying the ongoing equipment failures are accessible, management can access the data within one minute, which was the target.

3. Data visualization: Charts and visualization can be viewed from mobile and desktop devices and can be filtered according to user preferences.

4. Feature Functionality: All module features performed according to the predefined criteria, with the following evaluation results: Production Operators (C2: 5/5 and 6/6), Maintenance Supervisor (C3: 1/1), combined Maintenance Department (C1 and C3: 13/13), and Management Team (C4 – C7: 2/2). The Maintenance Technician (C1) modules, however, did not fully meet the intended purpose, scoring 7/8 and 7/9.

It should be noted that these results were obtained by the author rather than by end users. Therefore, while indicative of system performance, they cannot be considered fully representative of the CMMS's suitability for routine operational use by the client.

6.3 Discussion of results

The primary aim of this thesis was to investigate the feasibility of developing a maintenance management system based on a no-code platform. Four research questions were formulated to define the study's scope:

RQ1. What type of data should the maintenance system collect?

RQ2. How can the development of a maintenance system benefit from the axiomatic design approach?

RQ3. To what extent can a no-code platform be used to develop a CMMS by users without coding knowledge?

RQ4. In what ways does the no-code-based CMMS add value to the daily operations?

The literature review on maintenance theory and the client's existing maintenance process descriptions, combined with a qualitative approach involving observations of daily maintenance activities and personnel interviews, formed the theoretical framework for addressing the first research question (RQ1). The review highlighted the differences between various maintenance strategies. By analyzing the client's current and anticipated requirements, the necessary level of maintenance data was determined (Ch. 3.2 - Analyses).

The second research question (RQ2) examined the benefits of applying axiomatic design to the maintenance system. Axiomatic design facilitated the establishment of a holistic view of maintenance needs. Stakeholder requirements were translated into customer attributes (CAs), which in turn informed the functional requirements (FRs) and design parameters (DPs). These elements were systematically mapped in the ideal model (Ch. 4 – Maintenance System Development – Ideal Model), supporting the creation of a coherent system architecture that aligns with user needs (Figure 14).

The third research question (RQ3) addressed the use of a no-code platform for application development by users without programming knowledge. As described in Ch. 2.3.2 – Functionality of a No-Code Platform, AppSheet functions resemble Excel formulas, making prior experience with spreadsheets advantageous. Additionally, its visual editor, existing templates, and AI support facilitate context and behavior creation.

Based on the author's notes, it took approximately 80 hours in total for the author to study, build, and refine a functional CMMS using a no-code platform. The process included studying in the provider's learning center, watching tutorials, and exploring experiences shared in developer forums. It should also be noted that the software architecture had already been developed earlier (Ch. 4 – Maintenance System Development – Ideal Model), which reduced the need for a trial-and-error approach.

The fourth research question (RQ4) examined how a no-code-based CMMS can add value to daily operations. As discussed in Chapter 2.3.1, *From Low-Code to No-Code Applications*, key advantages include enhanced data privacy – achieved by reducing reliance on external developers and retaining operational data in-house – as well as time and cost savings, since end users can design and develop solutions tailored to their specific needs.

In this study, the no-code CMMS was accessible via mobile devices, enabling real-time data collection, automated system-based calculations, and continuous monitoring regardless of location. Where free-text fields were used, the availability of a speech-to-text function enabled efficient recording of maintenance data. This level of mobility supports location-independent work practices and can contribute to improved productivity.

The test results indicate that the system is capable of effectively capturing relevant operational data, thereby generating tangible value for daily operations. These outcomes are primarily underpinned by automated data collection processes, which enable data-driven decision-making and, consequently, contribute to enhanced operational efficiency.

7 Conclusion

The study showed that the data to be collected is dependent on customer needs and the environment. Those can be solved by using a systematic approach, which the axiomatic design approach can provide. By investigating the CAs, the FRs can be determined and subsequently by the experience of the designers, the DPs can be defined. Eventually, the system architecture can be formed based on the relationships between FRs and DPs.

The no-code platform offers a way to create software without coding knowledge and is therefore accessible to most users. However, without a proper systematic approach, data integration within the organization may be insufficient, and updating, refining, scaling, and troubleshooting may still require relevant coding skills.

The CMMS can automate manually performed daily tasks, thereby freeing resources and providing a single source of up-to-date data, reducing duplication. In this case, the possibility to use the application on a mobile device frees resources, as data can be stored in real-time and on-site.

Additionally, built features such as QR code scanning, NFC reading, and speech-to-text support a more streamlined process by reducing human error and improving traceability.

In conclusion, the no-code-based CMMS demonstrates strong potential for supporting maintenance management in SMEs, providing a practical, accessible, and adaptable solution for enhancing workflow efficiency, data collection, automated processing, and decision-making.

This thesis has not addressed the potential risks associated with concentrating multiple functions within a single software ecosystem, where the dependability of the service provider may become a critical factor as functions are scaled up. When implementing a new system, a thorough risk assessment should always be conducted.

References

2004/1016, 2021. *Laki eräiden teknisten laitteiden vaatimustenmukaisuudesta, 2004/1016.* [Online]
Available at: https://www.finlex.fi/fi/laki/ajantasa/2004/20041016
[Accessed 17 02 2025].

400/2008, 2011. *Valtioneuvoston asetus koneiden turvallisuudesta.* [Online]
Available at: https://www.finlex.fi/fi/laki/ajantasa/2008/20080400
[Accessed 18 02 2025].

403/2008, 2019. *403/2008 - Government Decree on the Safe Use and Inspection of Work Equipment.* [Online]
Available at: https://www.finlex.fi/fi/laki/ajantasa/2008/20080403
[Accessed 18 02 2025].

Alshakhshir, F. & Howell, M. T., 2021. *Data Driven Energy Centered Maintenance (2nd edition),* s.l.: River Publishers. Retrieved from
https://app.knovel.com/hotlink/pdf/id:kt012KWBC3/data-driven-energy-centered/basic-philosophy.

Amazon Web Services, Inc., 2025. *Generative AI-powered, low-code application builder - AWS App Studio.* [Online]
Available at: https://aws.amazon.com/appstudio/
[Accessed 2025 03 16].

Anderton, R., Botelho, V. & Reimers, P., 2023. *European Central Bank: The ECB Blog - More digital, more productive? Evidence from European firms.* [Online]
Available at:
https://www.ecb.europa.eu/press/blog/date/2023/html/ecb.blog230621~3c2f72aa70.en.html
[Accessed 22 10 2024].

Basson, M. & Moubray, J., 2018. *RCM3: Risk-Based Realibility Centered Maintenance.* South Norwalk, CT: Industrial Press, Inc..

Bodicherla, B., 2025. *"The Rise of Low-Code/No-Code Development: Democratizing Application Development", Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol, vol. 11, no. 2, pp. 171–178.* Ramnagar, Uttar Pradesh (U.P.), Shauryam Research Institute.

Brown, C. A., 2006. *ICAD-2016-15 Kinds of coupling and approaches to deal with them.* [Online]
Available at: https://www.axiomaticdesign.com/wp-content/uploads/icad2006_15.pdf
[Accessed 22 04 2025].

Calligaris, S. et al., 2023. *OECD: The slowdown in Finnish productivity growth - Causes and consequences.* [Online]
Available at: https://www.oecd.org/content/dam/oecd/en/publications/reports/2023/02/the-slowdown-in-finnish-productivity-growth_6ea7cde3/c1fad5b3-en.pdf
[Accessed 22 10 2024].

Forbes: A. Bridgwater, 2020.. *Google Says Yes To No-Code, Acquires AppSheet.* [Online]
Available at: https://www.forbes.com/sites/adrianbridgwater/2020/01/17/google-says-yes-to-no-code-acquires-appsheet/
[Accessed 25 06 2025].
Forrester Research: Richardson, C. et al., 2014. *New Development Platforms Emerge For Customer-Facing Applications - Firms Choose Low-Code Alternatives For Fast, Continuous, And Test-And-Learn Delivery.* [Online]
Available at: https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411
[Accessed 16 03 2025].
Geekwire: M.Nickelsburg, 2015. *Startup Spotlight: Appsheet lets non-developers build custom mobile apps.* [Online]
Available at: https://www.geekwire.com/2015/appsheet/
[Accessed 18 06 2025].
GeekWire: S.Taylor, 2020. *Google Cloud acquires AppSheet, a Seattle startup that sells no-code app development software.* [Online]
Available at: https://www.geekwire.com/2020/google-cloud-acquires-seattle-startup-appsheet-built-no-code-app-development-platform/
[Accessed 23 06 2025].
Google, 2024. *Appsheet.* [Online]
Available at: https://cloud.google.com/appsheet
[Accessed 14 09 2024].
Google, 2025.. *AppSheet - About.* [Online]
Available at: https://about.appsheet.com/home/
[Accessed 26 06 2025].
Google, 2025. *Google AppSheet | Build apps with no-code.* [Online]
Available at: https://about.appsheet.com/home/
[Accessed 16 03 2025].
Google, 2026. *AppSheet Help - Build expressions.* [Online]
Available at: https://support.google.com/appsheet/table/10104782?hl=en
[Accessed 6 4 2026].
Google, 2026. *AppSheet Help - Explore the app editor.* [Online]
Available at: https://support.google.com/appsheet/answer/12290157?hl=en
[Accessed 2026 4 6].
Google, 2026. *AppSheet pricing.* [Online]
Available at: https://about.appsheet.com/pricing/
[Accessed 6 4 2026].

Google, 2026. *Restrictions, limits, and known issues.* [Online]
Available at: https://support.google.com/appsheet/answer/12653576?hl=en&ref_topic=12632222
[Accessed 03 05 2026].

Hakanen, J. & Kaltiainen, J., 2024. *Finnish Institute of Occupational Health: Severe occupational burnout has increased.* [Online]
Available at: https://www.ttl.fi/en/topical/press-release/severe-occupational-burnout-has-increased
[Accessed 22 10 2024].

Hamasha, M. M. et al., 2023. *Strategical selection of maintenance type under different conditions.* London, Scientific Reports (Nature Publisher Group).

International Automotive Task Force, IATF, 2016. *IATF 16949:2016 - Automotive quality management system standard - Quality management system requirements for automotive production and relevant service parts organisation.* 1st ed. s.l.:s.n.

Järviö, J. & Lehtiö, T., 2012. *Kunnossapito - tuotanto-omaisuuden hoitaminen.* 5 ed. Helsinki: KP-Media Oy.

Koski, H. & Fornaro, P., 2024. *"Digitalization and Resilience: Data Assets and Firm Productivity Growth During the COVID-19 Pandemic" ETLA Working Papers No 113..* [Online]
Available at: http://pub.etla.fi/ETLA-Working-Papers-113.pdf
[Accessed 22 10 2024].

Lean Community: M. Bartosz, 2021. *MTBF MTTR MTTF: TPM Indicators.* [Online]
Available at: https://leancommunity.org/mtbf-mttr-and-mttf/
[Accessed 16 03 2025].

Luthra, S., Garg, D., Agarwal, A. & Mangla, S. K., 2021. *Total Quality Management (TQM): Principles, Methods, and Applications.* 1st ed. Boca Raton, FL: Taylor & Francis Group, LLC.

METSTA, Metalliteollisuuden Standardisointiyhdistys ry, 2020. *SFS-käsikirja 100-1:2020 - Koneiden turvallisuus. Osa 1: Suunnittelun perusteet ja riskin arviointi.* 2 ed. Helsinki: Suomen Standardisoimisliitto SFS ry.

Microsoft, 2024. *What is a low-code development platform?.* [Online]
Available at: https://www.microsoft.com/en-us/power-platform/products/power-apps/topics/low-code-no-code/low-code-platform
[Accessed 14 09 2024].

Microsoft, 2025. *Microsoft Power Apps - kehitä sovelluksia tekoälyn avulla.* [Online]
Available at: https://www.microsoft.com/fi-fi/power-platform/products/power-apps
[Accessed 2025 03 16].

Nakajima, S., 1988. *Introduction to TPM: Total Productive Maintenance.* Portland, OR: Productivity Press.

Nguyen, D. Q., Brammer, C. & Bagajewicz, M., 2008. *New Tool for the Evaluation of the Scheduling of Preventive Maintenance for Chemical Process Plants.* Norman, OK, American Chemical Society.

Nowlan, F. S. & Heap, H. F., 1978. *Reliability-Centered Maintenance,* Springfield, VA: U.S. Department of Commerce - National Technical Information Service.

Occupational Safety and Health Administration in Finland, 2023. *Manufacturer's responsibilities.* [Online]
Available at: https://tyosuojelu.fi/en/market-surveillance/machinery-and-equipment/manufacturer
[Accessed 18 02 2025].

OECD, 2024. *"Productivity and unit labour cost by industry, ISIC Rev. 4", OECD Productivity Statistics (database).* [Online]
Available at: https://doi.org/10.1787/data-00687-en
[Accessed 14 09 2024].

Official Journal of the European Union, 2019. *Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on machinery, and amending Directive 95/16/EC (recast).* [Online]
Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02006L0042-20190726
[Accessed 17 02 2025].

Pačaiová, H. & Ižaríková, G., 2019. *"Base Principles and Practices for Implementation of Total Productive Maintenance in Automotive Industry" Kvalita Inovácia Prosperita, vol. 23, no. 1, 2019, pp. 45–59.* [Online]
Available at: https://doi.org/10.12776/QIP.V23I1.1203
[Accessed 22 10 2024].

Sanchis, R., García-Perales, Ó., Fraile, F. & Poler, R., 2019. *Low-Code as Enabler of Digital Transformation in Manufacturing Industry.* Basel, MDPI, Applied Sciences, [Online] 10(1), 12. https://doi.org/10.3390/app10010012.

Siirilä, T., 2008. *KONETURVALLISUUS - EU:n direktiivien ja standardien soveltaminen käytännössä.* 2 ed. Keuruu: Inspecta Koulutus Oy.

Stevenson, W. J., 2021. *Operations Management.* 14 ed. New York, NY: McGraw-Hill Education.

Stevenson, W. J. & Kull, T. J., 2024. *Operations and Supply Chain Management.* 15 ed. New York, NyY: McGraw Hill LLC.

Suh, N. P., 2001. *Axiomatic Design - Advances and Applications.* 1 ed. New York: Oxford University Press.

Tongdan, J., 2019. *Reliability Engineering and Services.* 1st ed. Hoboken, NJ: John Wiley & Sons Ltd.

Tuominen, K., 2010. *Tehoa ja laatua kunnossapidon kehittämiseen: Total Productive Maintenance: Mikä erottaa menestyjät keskinkertaisista?.* 1st ed. Jyväskylä: Readme.fi.

Viljoen, A., Hein, A. & Krcmar, H., 2026. *Low-code development platform ecosystems. Electronic Markets 36:12; https://doi.org/10.1007/s12525-025-00848-x,* s.l.: Springer Nature.