Project number: 826276

CPS4EU

Cyber Physical Systems for Europe

D8.7 – Test and Validation plan

Reviewer (name – company):

Bernhard Bauer (UNIVERSITAET AUGSBURG)

Eric Dujardin (THALES)

Dissemination level: Public

Version	Date	Author (name – company)	Comments
0	4/2/2022	Alexander Mages (TRUMPF) David Pluntze (WIKA) G. Zappalà, A.Pertotti, G. Caminale, N. Rebella, (Leonardo)	For internal review and PCO & TPM review
1	4/3/2022	Alexander Mages (TRUMPF) David Pluntze (WIKA) N. Rebella, (Leonardo)	First issue

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Table of content

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

1. INTRODUCTION

1.1. Purpose

This document is related to Task 8.4 of WP8 concerning the Test and Validation of the prototypes of industrial use cases.

In task T8.1 the use case requirements of the industrial use cases in CPS4EU have been elicited and established, as captured in deliverable D8.9.

In task T8.2 those use cases were analysed to produce the use case model and high-level design. Deliverable D8.4 describes the use case components that are envisaged to satisfy use case needs, how they work together, and the components where CPS4EU modules/PI-ARCHs are used.

In Task 8.3 the use case components are implemented to produce a prototype of the CPS according to the use case design in Task 8.2 to address the requirements identified in T8.1.

Task T8.4 deals with the verification and validation of those prototypes. Verification and validation (also abbreviated as V&V) are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfils its intended purpose.

This document provides a description of the general strategy and the details of the tests for verification and validation of the prototypes implemented in WP8. In general use case prototypes are tested and validated against the user requirements established in D8.9.

Tests and validation results will be reported in D8.8.

1.2. Scope

The following WP8 Industry Automation Use Cases are addressed:

- UC4 - Automatic Vacuum System (LEONARDO)
- UC5 - Trimming Quality Improvement (LEONARDO)
- UC7 - Aircrafts Health Management System (LEONARDO)
- UC8 - Material Flow Analytics and Simulation (TRUMPF)
- UC9 - Mobile CPSs (WIKA)

The use case Thermoplastic Production Line Monitoring (LEONARDO) is not included as the implementation of a prototype of that use case is outside the scope of the project.

1.3. Document structure

The first chapter describes the different steps of the test plan in general terms to lay the basis of a common language.

Next chapters are dedicated to descriptions of the testing and validation plans of each use case where the following aspects are addressed:

- The high-level description and objective of the use case
- The strategy adopted for the test and validation steps, adapted to the specificities of the use case, the context and the partners' know-how and tools;
- The detailed test plan, showing the list of test cases that are planned

As an annex, you can find the description of each test case that is planned.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

1.4. Link to other documents/tasks

ID	Description
D.8.4	Use design and modelling v2
D.8.9	Use case requirements v3

1.5. Definitions, acronyms, and abbreviations

Acronym / abbreviation	Description
AHMS	Aircraft Health Management System
AOCP	Aircraft out of commission for parts
AOG	Aircraft on ground
Blob	Binary Large Object
CPS	Cyber Physical System
CRISP-DM	CRoss Industry Standard Process for Data Mining
GF	Ground Framework
ML	Machine Learning
MTBUR	Mean Time Between Unscheduled removals
MQTT	Message Queuing Telemetry Transport
NTP	Network Time Protocol
OPC-UA	Open Platform Communications Unified Architecture
PI-ARCH	Pre-integrated Architecture
URR	Unscheduled removal rate
UWB	Utra Wide Band

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

2. TEST AND VALIDATION GENERAL DEFINITIONS

Test and validation naming and contents can vary significantly from one domain to another and between partners. Hence, we define in this section general test and validation steps that can be adapted to each use case but provide a common ground to facilitate understanding across the CPS4EU project.

2.1. Requirements validation

Requirement validation consists in checking the consistency of your requirements with a specific method or tool. It is not strictly speaking part of the test plan but it allows early detection of problems, at the conception stage.

2.2. Unit testing

Software unit tests are typically automated tests written and run by developers to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software, e.g. a function, a procedure or a method depending on programming style. There is generally no bug report associated with these tests as problems are fixed as they are found.

In this document only the methods and tools used for unit testing are mentioned, but not the specific tests made.

2.3. Component testing

Component testing is the verification of the individual functionalities and capabilities expressed by that component, related to the initial requirements and the "common" or "domain specific" characteristics attributed to the individual component of the system. A component is considered validated when it meets the expected functional and performance requirements that are implemented by that component, according to the system design.

In this document component test plans are described only for those components that play a relevant role in system validation and when specific user requirements are met in a specific component.

2.4. Integration testing

Integration tests are aimed to verify the interactions between units or components, should they be software or hardware and test them as a group. They are intended to check the interfaces/exchanges between two components.

Integration tests between important parts of the overall system (e.g. two components or two stand-alone software codes from different providers) should be described in this document. Otherwise, integration is implicitly checked at the system level, where all the system components are tested together.

2.5. System testing

System verification tests a completely integrated system to verify that the system meets its specified requirements, with a black-box approach in a production-like environment. Tests cases are usually derived from the system requirements.

This document includes an overview of the system verification test plan.

2.6. System acceptance testing

Acceptance tests evaluate that the system is compliant with the business requirements and assess whether it is acceptable for delivery and production. Acceptance criteria are defined to enable the user, customers or other

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

authorized entities to determine whether to accept the system. Their designation may differ depending on the industry, the domain or the context (Factory acceptance Tests, beta test …). This may also include a site validation period, where the system operates in real conditions, or in parallel with a pre-existing system.

As use case prototypes developed in WP8 are mainly a proof of concept and do not deliver a final industrial product, they are validated against the user requirements and accepted if the use case reached its goal.

This document includes a full description of acceptance tests plan where applicable.

3. TEST AND VALIDATION PLAN UC4 - AUTOMATIC VACUUM SYSTEM (LEONARDO)

3.1. Background of the use case

The use case deals with a specific assembly process on large composite structures and aims to automate drilling activities on such structures that currently are human driven.

During drilling activities, the human intervention is twofold: one person drills while the other – positioned on the opposite side of the large structure – has to vacuum the carbon fibre dust that is produced by drilling. The use case will automate the movements of the vacuum system to “follow” the drill position.

The objective of this use case is to move the vacuum automatically to precisely follow the position of the DRILL to vacuum the carbon fibre dust without manual intervention.

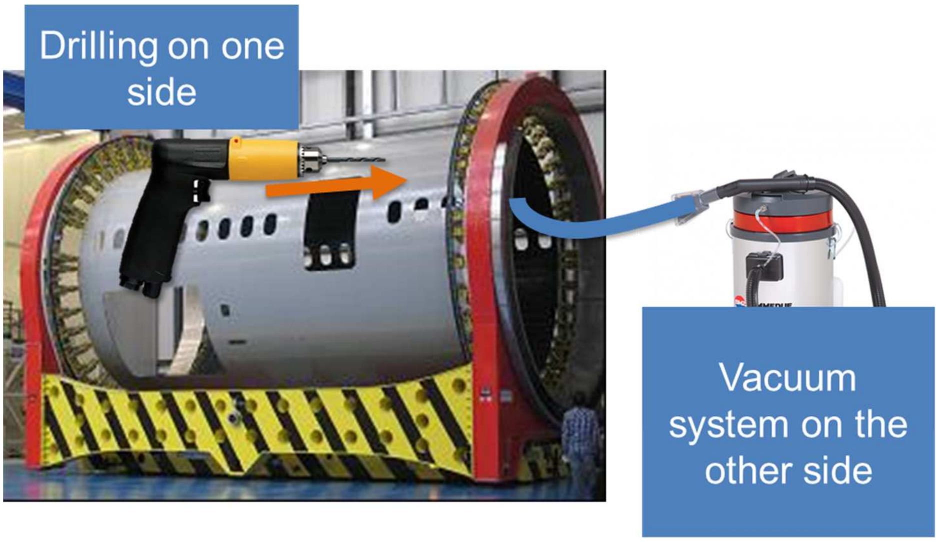


Figure 1 - UC4 overview.

More information on the background and use case requirements can be found in D8.9.

3.2. Strategy for test and validation

3.2.1. High-level view of the strategy

The use case goal is not to build a final product but to have a prototype to demonstrate the proposed solution and selected technology in a real environment (TRL7 prototype).

The adopted test and validation strategy encompasses several test steps: from the unit testing and component testing, through integration testing and system testing up to the acceptance testing aimed to validate the technological prototype in Leonardo production environment on the plant in Grottaglie (Taranto) where the drilling of fuselage sections is carried out as part of the aircraft fuselage production process.

Unit and component testing are performed as part of the development by the partner responsible to implement a specific component of the use case final prototype.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

The use case prototype is made of different components as described in D8.4, encompassing architectures and technological modules developed in CPS4EU and specific components:

- the Drill add-on that includes several modules for drill tip proximity detection, localization, interaction with the drill operator;
- a gateway based on an industrial computing platform with an IoT integration framework, as a field interconnection module that hosts the control logic and supports the communication between the drill and the vacuum;
- a vacuum positioning system based on a cobot that moves the vacuum hose with its anthropomorphic arm
- an enterprise data analysis platform where the main events of the drilling process are collected, which exposes the Monitoring interface that shows the progress of the process.

Each partner that is responsible to deliver a component shall separately prove that the component meets the use case specifications and that it can work with the components it has to be integrated with. When a component is delivered it is assumed to have passed the internal testing by the supplier partner to check those aspects.

To that purpose stubs and emulators are adopted by each partner to simulate at their labs other components and the load/interactions of the use case scenario.

When all components are ready, they are assembled and tested for integration with the other prototype components and for verification of the complete system. Final Integration tests take place directly on the plant in Grottaglie to ensure the prototype fits the work environment and meets the working conditions and operational constraints of the production process. The interactions between the components of the prototype are tested, checking separately each step of the sequence of the process and that every component receives and/or produces the expected messages according to the external interface specifications of the planned interactions.

Acceptance tests are finally executed on the plant by the end users to validate the system behaviour against the usage patterns envisaged to fulfil the user needs.

3.2.2. Tools and resources

The execution of unit tests and full component testing is part of the development process. The partner that supplies a component is responsible to execute those tests at their labs (workbench testing), before delivering the component for integration into the system. Unit tests are usually executed automatically at every build. Component testing as well as integration, system and acceptance testing involve manual activities to setup and execute the test cases.

Emulation tools are envisaged to allow the test of each component separately and to support incremental integration testing between components before the full prototype is in place. They include:

- Stubs that read data from input files and produce the MQTT packets originated from the drill add-on;
- The MQTT broker to simulate the MQTT topics where messages have to be published;
- The Gateway platform with Eclipse Kura to emulate the edge computing and connectivity PI-ARCH before it is ready;
- Emulation on Azure Cloud of the data analysis platform.

The execution of the test cases in this test plan is managed in Azure DevOps, a tool for Agile development that allows to share information between all actors of the project team. Azure DevOps is a team workspace that allows to organize and share work, so every team member has access to the information they need. In Azure DevOps the use case requirements are captured as features and the expected system behaviour as user stories. Members of the team can ask questions or make remarks that are directly integrated into the requirements. Test cases can also be tracked as work items and organized in test plans for their execution. Bugs or issues raised from the test execution can also be managed.

3.2.3. Tests definition

The tests in the plan have been set to check whether the prototype satisfies the elicited use case requirements (see D8.9) where means of validation is “prototype” or “preliminary prototype” and demonstrate that the CPS

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

system is able to achieve the use case goal as captured by the metrics of the technological KPIs 3.1 for this use case:

- vacuum positioning is automated;
- vacuum position matches the drill position;
- post production services are enabled: information for both vacuum and drill processing is provided for post-production analysis.

The tests in the plan are mainly focused on the functional use case requirements checked at the system level and acceptance level. A few component tests are included to prove requirements concerning features implemented at the component level. Acceptance tests cover the intended usage of the system by the final users.

As for the non-functional requirements:

- Design requirements are mainly hardware/software requirements and have been taken into account in the components and system design (see D8.4 for the design choices made to meet those requirements);
- Security requirements are not tested per se but are considered in the design of the system and setup of the IT infrastructure where the system is deployed (see D8.4).
- Performance tests are limited to check the prototype response and precision in positioning the vacuum automatically. More extensive testing goes beyond the scope of this preliminary prototype;
- safety, operational and usability requirements should be addressed in final product and tests are not covered in this test plan;
- Cyber-security is handled at conception level by using hardened operating system, network, communication and system security measures according to Leonardo IT security policy. At the implementation stage, specific tests are dedicated to vulnerability issues (DevSecOps).

The table below lists the test cases that are planned to validate the requirements at the different stages of the implementation and prototyping process. A reference to the use case requirements that the use case is intended to test can be found. It also shows the link to the detailed test definition that can be found in annex 8.1.

Test name	Test level	Test ref.	Req.ID	Comment
Vacuum positioning	Component	8.1.1	UC4-FNC-02 UC4-FNC-03	
Vacuum coverage of fuselage area	Component	8.1.2	UC4-FNC-01	
Drill localization	Component	8.1.3	UC4-FNC-03	
Drill close to the fuselage	Integration	8.1.4	UC4-FNC-04	
Operator ready to drill at the position	Integration	8.1.5	UC4-FNC-04	
Vacuum positioning command	Integration	8.1.7	UC4-FNC-04	
Vacuum positioning feedback	Integration	8.1.8	UC4-FNC-04	
Consensus to drill	Integration	8.1.9	UC4-FNC-04	

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Operator Consensus for drilling (negative)	System	8.1.10	UC4-FNC-05	
Operator Consensus for drilling (positive)	System	8.1.11	UC4-FNC-05	
Vacuum consensus for drilling (positive)	System	8.1.12	UC4-FNC-06	
Vacuum consensus for drilling (negative)	System	8.1.13	UC4-FNC-06	
Drill tip wear estimation (good tip)	Component	8.1.14	UC4-FNC-07	
Drill tip wear estimation (worn out tip)	Component	8.1.15	UC4-FNC-07	
Dynamic Obstacle perception	Component	8.1.16	UC4-FNC-08	
Static Obstacle perception	Component	8.1.17	UC4-FNC-08	
Vacuum Positioning time	System	8.1.18	UC4-PRF-01	
Drilling process cycle	System/Acceptance	8.1.19		

Table 1 – Test definition.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

4. TEST AND VALIDATION PLAN UC5 - TRIMMING QUALITY IMPROVEMENT (LEONARDO)

4.1. Background of the use case

During trimming/milling activities delamination can be experienced on parts, caused by different phenomena that are difficult to be managed because of the high complexity and high numbers of variables (vibration, detachment of the part being cut, tool wear, speed, humidity, temperature, air pressure, etc.).

The objective of this use case is to collect data coming from sensors and numerical control machines (CNC), analyse them with a quality statistics algorithm and understand the main root causes of defects and then provide real-time information in order to change the setting of machine parameters to reduce the risk of damage or defect.

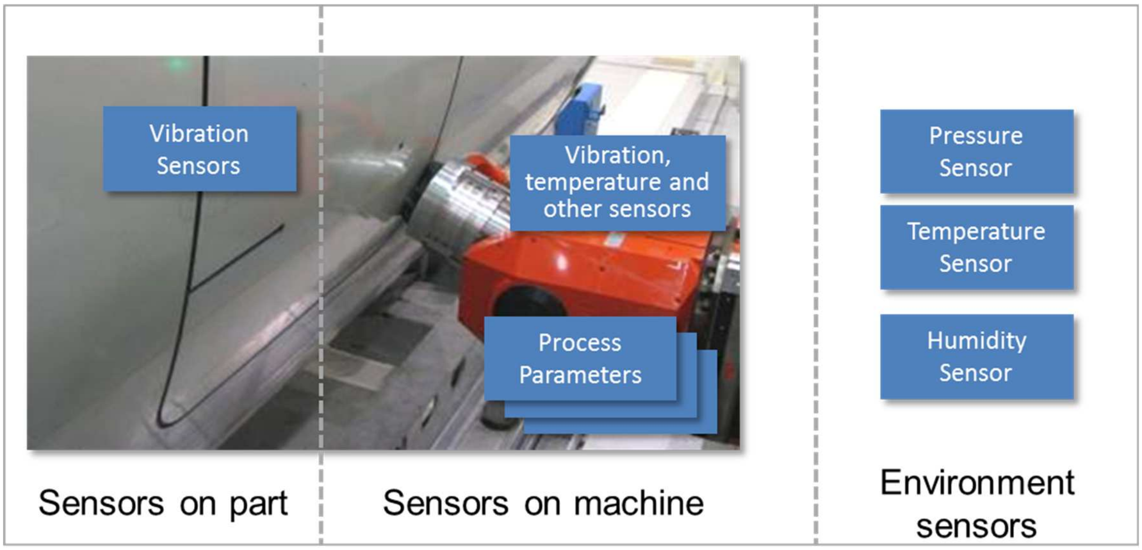


Figure 2 - UC5 overview.

More information on the background and use case requirements can be found in D8.9

4.2. Strategy for test and validation

4.2.1. High-level view of the strategy

The use case goal is not to build a final product but to have a prototype to demonstrate the proposed solution and selected technology in a real environment (TRL7 prototype).

The adopted test and validation strategy encompasses several test steps: unit testing, component testing, integration testing up to the system testing and acceptance testing aimed to prove the technological prototype in Leonardo production environment of the plant in Grottaglie (Taranto).

Unit and component testing are performed as part of the development by the partner responsible to implement a specific component of the use case final prototype.

The use case prototype is made of different components as described in D8.4, encompassing architectures and technological modules developed in CPS4EU and specific components:

- a distributed sensing layer with sensing nodes for the various sources of process variables responsible for turning sensor signals into a time series of data for the relevant process variable, with samples at the relevant frequency;
- a gateway based on an Industrial computing platform with an IoT integration framework, responsible for collecting the data streams from the distributed sensing nodes and of sending them to the remote enterprise data analysis platform; it also runs the prediction model on the edge;

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

- an enterprise data analysis platform that supports the data scientist in analysing the process data collected from the plant to discover correlations and in producing a prediction model using machine learning techniques.
- an HMI that shows the operator the alerts raised in real-time by the prediction model

Each partner that is responsible to deliver a component shall separately prove their component meets the use case specifications and can work with the components it has to be integrated with. When a component is delivered it is assumed to have passed the internal testing by the supplier partner to check those aspects.

To that purpose stubs and emulators are adopted by each partner to simulate at their labs other components and the load/interactions of the use case scenario.

When all components are ready, they are assembled and tested for integration with the other prototype components and for off-line verification of the complete system behaviour before it is applied on the production process.

Integration tests will take place directly on the plant in Grottaglie to ensure the prototype fits the work environment and meets the working conditions and operational constraints without interfering with the normal production process. The complete data acquisition chain is tested, checking separately each step of the data flow and that every component receives and/or produces the expected data according to the external interface specifications of the data exchange.

After system verification, the prototype of the data acquisition chain will be used during trimming production processes of aircraft fuselages, to obtain enough data to train a prediction model that is able to correlate the process variables with the quality of the production output. According to the CRISP-DM methodology the prediction model will be trained using part of the available data (80%) and validated against the rest of those data (20%) to check it produces the expected output when experimented on data it was not trained on before.

Acceptance tests are finally executed on the plant by the end users to validate the system behaviour against the usage patterns envisaged to fulfil the user needs.

4.2.2. Tools and resources

The execution of unit tests and full component testing is part of the development process. The partner that supplies a component is responsible to execute those tests at their labs (workbench testing), before delivering the component for integration into the system. Unit tests are usually executed automatically at every build. Component testing as well as integration, system and acceptance testing involve manual activities to setup and execute the test cases.

Emulation tools are envisaged to allow the test of each component separately and to support incremental integration testing between components before the full chain of the final prototype is in place. They include:

- Stubs that read data from input files and produce the MQTT data stream packets originated from sensor nodes;
- MQTT broker to simulate the MQTT topics where data have to be published;
- Gateway platform equipped with Eclipse Kura to emulate the edge computing and connectivity PI-ARCH before it is ready;
- Emulation of the data analysis platform on Azure Cloud.

Additional auxiliary tools are required to simulate the use case scenario e.g. vibrations generator, air blower.

The execution of the test cases in this test plan is managed in Azure DevOps, a tool for Agile development that allows to share information between all actors of the project team. Azure DevOps is a team workspace that allows to organize and share work, so every team member has access to the information they need. In Azure DevOps the use case requirements are captured as features and the expected system behaviour as user stories. Members of the team can ask questions or make remarks that are directly integrated into the requirements. Test cases can also be tracked as work items and organized in test plans for their execution. Bugs or issues raised from the test execution can also be managed.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

4.2.3. Tests definition

The tests in the plan have been set to check the prototype satisfies the elicited use case requirements (see D8.9 where means of validation is “prototype” or “preliminary prototype”) and demonstrate that the CPS system is able to achieve the use case goal as captured by the metrics of the technological KPIs 3.1 for this use case:

- enable the data collection of trimming process variables from different sources;
- enable the analysis of the collected data with machine learning and statistics algorithms;
- enable defects prediction.

The tests in the plan are mainly focused on the functional use case requirements checked at the system level and acceptance level. A few component tests are included as a proof for requirements concerning features implemented at component level. Acceptance tests cover the intended usage of the system by the final users.

As for the non-functional requirements:

- Design requirements are mainly hardware/software requirements and have been taken into account in the components and system design (see D8.4 for the design choices made to meet those requirements);
- Security requirements are not tested per se but are considered in the design of the system and setup of the IT infrastructure where the system is deployed (see D8.4).
- Performance and load tests are minimal and limited to check the prototype is able to cope with the expected data rates and precision. More extensive testing goes beyond the scope of this preliminary prototype and is not covered;
- safety, operational and usability requirements should be addressed in final product and tests are not included in this test plan
- Cyber-security is handled at conception level by using hardened operating system, network, communication and system security measures according to Leonardo IT security policy. At the implementation stage, specific tests are dedicated to vulnerability issues (DevSecOps).

The table below lists the test cases that are planned to validate the requirements at the different stages of the implementation and prototyping process. A reference to the use case requirements that the use case is intended to test can be found. It also shows the link to the detailed test definition that can be found in annex 8.2.

Test name	Test level	Test ref.	Req.ID	Comment
Environment parameters measurement	Component	8.2.1	UC5-FNC-01	
Collection on the field of environment parameters measures	Integration	8.2.2	UC5-FNC-06	
Worked part vibration measurement	Component	8.2.3	UC5-FNC-02	
Collection on the field of the worked part vibration measure	Integration	8.2.4	UC5-FNC-06	
Trimming head vibration measurement	Component	8.2.5	UC5-FNC-02	
Collection on the field of the Trimming head vibration measure	Integration	8.2.6	UC5-FNC-06	

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Trimming tool tip vibration measurement	Component	8.2.7	UC5-FNC-02	
Collection on the field of the Trimming tool tip vibration measure	Integration	8.2.8	UC5-FNC-06	
Trimming air flow measurement	Component	8.2.9	UC5-FNC-02	
Collection on the field of the Trimming air flow measure	Integration	8.2.10	UC5-FNC-06	
Trimming machine work parameters acquisition	Component	8.2.11	UC5-FNC-02	
Collection of Trimming machine work parameters	Integration	8.2.12	UC5-FNC-06	
Communication of collected measures to the enterprise data analysis platform	Integration	8.2.13	UC5-FNC-06	
Data staging of measures collected from the field	Component	8.2.14	UC5-FNC-06	
Data collection of relevant trimming process parameters from the field	System / Acceptance	8.2.15	UC5-FNC-06	
Data Loading of quality inspection data	Component	8.2.16	UC5-FNC-06	
Data analysis	Component/ System / Acceptance	8.2.17	UC5-FNC-06	
Prediction model validation	System / Acceptance	8.2.18	UC5-FNC-06 UC5-FNC-09	
HMI display	Component	8.2.19	UC5-FNC-07	
Communication of the collected measures and prediction output to the HMI	Integration	8.2.20	UC5-FNC-07 UC5-FNC-08	
Trimming process monitoring	System / Acceptance	8.2.21	UC5-FNC-07 UC5-FNC-08	
Real-time execution of the prediction model	System	8.2.22	UC5-PRF-01	

Table 2 – Test definition.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

5. TEST AND VALIDATION PLAN UC7 - AIRCRAFTS HEALTH MANAGEMENT SYSTEM (LEONARDO)

5.1. Background of the use case

The Aircraft Health Management System (AHMS) is devoted to gathering, collecting and analysing data concerning aircraft fleet maintenance.

The overall system (depicted in the figure below) consists of different modules, located both on-board and on-ground, providing data and HW / SW framework.

The objective is to collect and correlate data from the aircraft (failures, removed items and performance data), warehouse and other sources (knowledge base, manuals) to support AHMS users in:

- failure troubleshooting (Maintenance Operators);
- monitoring aircraft systems performance and anticipating possible failures (Department Engineers);
- procurement decisions, anticipating spare parts demand (Logistic Operators).

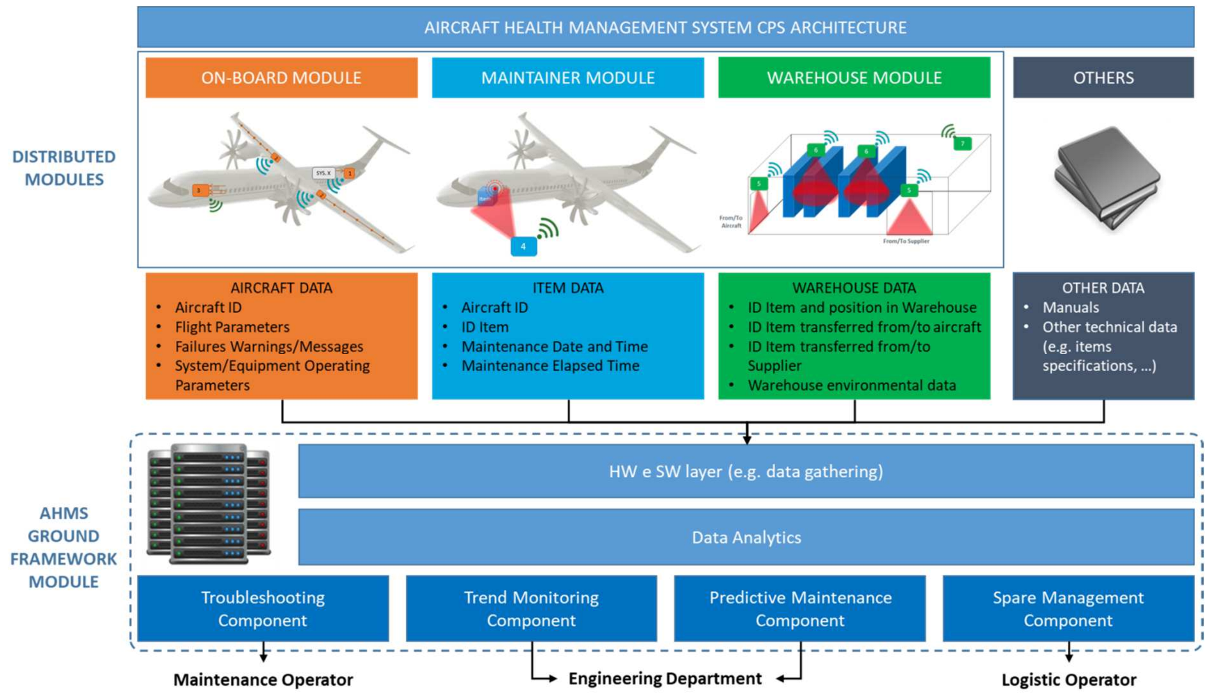


Figure 3 - AHMS CPS – overall picture

More information on the background and use case requirements can be found in D8.9

5.2. Strategy for test and validation

5.2.1. High-level view of the strategy

The use case goal is to demonstrate that data analytics and machine learning techniques can be used to support the decisions that Maintenance and Logistic operators have to take, with information obtained from intelligence on aircraft, maintenance and spare parts data that goes beyond what can be found directly from troubleshooting procedures, scheduled maintenance plans and spare parts availability.

Therefore, the focus of the use case prototype implemented in CPS4EU is on the cyber part of the CPS and the scope is limited to the ground framework component of the use case where aircraft data are collected and analysed, particularly to the Troubleshooting and Spare Management components of the ground framework that are implemented on a data analysis platform based on Azure technology (see the D8.4 "Use case

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

modelling and design v2”). The objective is not to have a full prototype of the final product but only to have a preliminary prototype to prove the data intelligence technology on a simulation of the real environment (TRL6).

The “physical” components of the CPS (i.e. aircraft on-board module, maintainer modules, warehouse module) are simulated as source files for the data that are originated in those components (i.e. aircraft data, removed items, spare parts data). In the use case prototype, the data gathering component that is responsible to collect and load those data about the physical process into the data analysis platform (see D8.4) is implemented to read the data from those source files. Therefore, integration testing aims at checking that data are taken and loaded from the source files. Unit testing is performed as part of the development.

The focus of the test and validation strategy is on Component testing to verify that the features implemented in the Troubleshooting and Spare Management components satisfy the use case requirements for those components as captured in D8.9 “Use case requirements v3”. For this use case, component testing overlaps with system testing, as the two components are separately responsible to implement a set of user requirements.

Acceptance tests are finally executed by the end users of the aircraft division, to validate the system behaviour against the usage patterns envisaged to fulfil the users’ needs.

5.2.2. Tools and resources

The execution of the test cases in this test plan is managed in Azure DevOps, a tool for Agile development that allows to share information between all actors of the project team. Azure DevOps is a team workspace that allows to organize and share work, so every team member has access to the information they need. In Azure DevOps the use case requirements are captured as features and the expected system behaviour as user stories. Members of the team can ask questions or make remarks that are directly integrated into the requirements. Test cases can also be tracked as work items and organized in test plans for their execution. Bugs or issues raised from the test execution can also be managed.

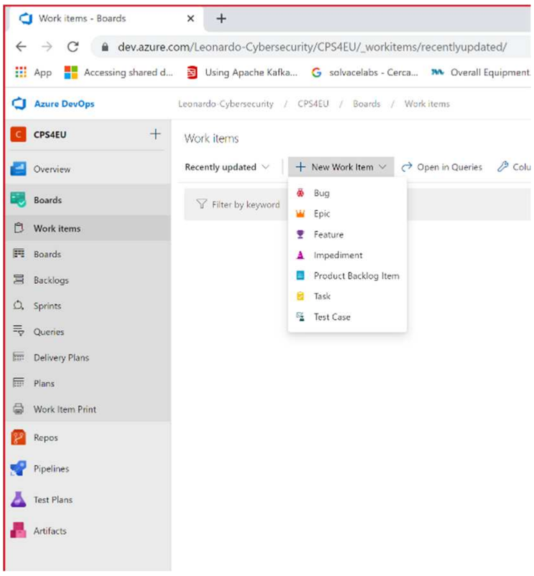


Figure 4 – Azure DevOps

Unit tests are executed automatically at every build of a software release. Component testing as well as integration and acceptance testing involve manual activities to setup and execute the test cases.

The execution of the test cases in the test plan, requires to setup data files as data sources for a set of data on the aircrafts to be analysed and covering their usage over a relevant stretch of time:

- aircraft flight parameters (including actual flight hours);
- aircraft equipment alerts and failures
- aircraft items removals
- warehouse stock data
- tracking of in/out parts

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Additional resources required to simulate the use case scenario are reference maintenance data on aircraft parts, foreseen flight hours, spare parts supply lead time, as well as other configuration information e.g. failures catalogue, material required for maintenance intervention. For test purposes that information is also simulated on file sources.

5.2.3. Tests definition

The tests in the plan have been set to check the prototype satisfies the elicited use case requirements (see D8.9) where means of validation is "preliminary prototype", and demonstrate that the CPS system is able to achieve the use case goal as captured by the metrics of the technological KPIs 3.1 for this use case:

- AHMS Troubleshooting component fulfils its requirements;
- AHMS Spare Management component fulfils its requirements;
- enable the collection of relevant data from different sources (aircraft failures, aircraft flight parameters, item removals, troubleshooting manuals, warehouse in/out tracking);
- enable data analysis in AHMS with statistics and machine learning algorithms;
- enable the identification of valid correlations about aircraft failures.

Therefore, the tests in the plan are focused on the functional use case requirements checked at the component/system level and acceptance level. At component level tests are aimed at verifying that the features implemented in the Troubleshooting and Spare Management components satisfy the use case requirements for those components (see D8.9), particularly the requirements that have high priority. Acceptance tests cover the intended patterns of usage to support the final users (maintenance operator, logistic operator, airframer) in their job.

Design requirements are mainly hardware/software requirements and have been taken into account in the components and system design and/or satisfied directly by the features of the Azure Stack software platform adopted to implement the Troubleshooting and Spare management components.

Performance and load tests are not covered as this is a preliminary functional prototype implemented on limited hardware resources and tested on a limited amount of data; the adopted Azure Stack platform supports horizontal and vertical scalability to cope with large amount of data.

Testing of security requirements is limited to check the prototype is able to manage different user roles and grant access according to the role of the user logged in. Extensive security testing goes beyond the scope of this preliminary prototype and should be addressed in a final product intended to be used by users of different customers. For the purpose of this preliminary prototype the data collected and loaded in the system are anonymized.

Cyber-security is handled at the implementation stage, where specific tests are dedicated to vulnerability issues (DevSecOps).

The table below lists the test cases that are planned to validate the requirements at the different stages of the implementation and prototyping process. A reference to the use case requirements that the use case is intended to test can be found. It also shows the link to the detailed test definition that can be found in annex 0.

Test name	Test level	Test ref.	Req.ID	Comment
Loading of aircraft failures data	Integration/Component	8.3.1	UC7-FNC-73	
Loading of aircraft flight parameters	Integration/Component	8.3.2	UC7-FNC-73	
Loading of items removals	Integration/Component	8.3.3	UC7-FNC-74	

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Loading of troubleshooting manuals	Integration/Component	8.3.4	UC7-DSG-17	
Troubleshooting component access by Maintenance Operator	System/Component	8.3.5	UC7-OPR-11	
Troubleshooting component access by Airframer Operator	System/Component	8.3.6	UC7-OPR-12	
List of fault events and event selection	System/Component	8.3.7	UC7-FNC-80 UC7-FNC-81 UC7-FNC-95	
Flight parameters chart	System/Component	8.3.8	UC7-FNC-83 UC7-FNC-84	
List of possible solutions for fault event	System/Component	8.3.9	UC7-FNC-76 UC7-FNC-77 UC7-FNC-78 UC7-FNC-79 UC7-FNC-82 UC7-FNC-85 UC7-FNC-86 UC7-FNC-87 UC7-FNC-91 UC7-FNC-97 UC7-FNC-98 UC7-DSG-15	
Maintenance Operator Notes	System/Component	8.3.10	UC7-FNC-90	
Airframer Notes	System/Component	8.3.11	UC7-FNC-89 UC7-FNC-92	
Association between Failures and Removals	System/Component	8.3.12	UC7-FNC-93	
Correlation analysis between failures and flight parameters	System/Component	8.3.13	UC7-FNC-94	
Calculation of Investigation Statistics	System/Component	8.3.14	UC7-FNC-96	
Maintenance Statistics visualization	System/Component	8.3.15	UC7-FNC-102 UC7-FNC-103 UC7-FNC-106	

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Export of maintenance activities report	System/Component	8.3.16	UC7-FNC-108	
Export of flight debrief report	System/Component	8.3.17	UC7-FNC-107	
Troubleshooting	Acceptance	8.3.18	-	
Investigation Data	Acceptance	8.3.19	-	
Analytics	Acceptance	8.3.20	-	
Troubleshooting Optimization	Acceptance	8.3.21	-	
Identification of Valid Correlations	Acceptance	8.3.22	-	

Table 3 – AHMS Tests definition (Troubleshooting)

Test name	Test level	Test ref.	Req.ID	Comment
Loading of warehouse data	Integration/Component	8.3.23	UC7-FNC-113	
Loading of warehouse in/out tracking data	Integration/Component	8.3.24	UC7-FNC-116	
Loading of Flight activity data	Integration/Component	8.3.25	UC7-FNC-115	
Spare Management component access by Logistic Operator	System/Component	8.3.26	UC7-OPR-11	
Spare Management component access by Airframer Operator	System/Component	8.3.27	UC7-OPR-12	
Scheduled maintenance activities expiration date	System/Component	8.3.28	UC7-FNC-114 UC7-FNC-120 UC7-FNC-121 UC7-FNC-141	
Reliability indicators	System/Component	8.3.29	UC7-FNC-119 UC7-FNC-122 UC7-FNC-123 UC7-FNC-124 UC7-FNC-125 UC7-FNC-141	
Updating of Reliability KPI	System/Component	8.3.30	UC7-FNC-126	

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Availability Warning and relevant performance indicators	System/Component	8.3.31	UC7-FNC-127 UC7-FNC-128 UC7-FNC-129 UC7-FNC-130 UC7-FNC-134	
Modification of weights and thresholds	System/Component	8.3.32	UC7-FNC-131	
Register AOG events	System/Component	8.3.33	UC7-FNC-132	
Recommendation of weights and thresholds	System/Component	8.3.34	UC7-FNC-133	
Visualization of performance indicators	System/Component	8.3.35	UC7-FNC-135 UC7-DSG-17 UC7-DSG-18 UC7-DSG-19	
Recommended stock size	System/Component	8.3.36	UC7-FNC-136	
Export of Parts availability report	System/Component	8.3.37	UC7-FNC-138	
Export of scheduled activities report	System/Component	8.3.38	UC7-FNC-139	
Export of top unreliable items report	System/Component	8.3.39	UC7-FNC-140	
Stock demand	Acceptance	8.3.40	-	
Top Unreliable Items monitoring	Acceptance	8.3.41	-	
Scheduled Maintenance monitoring	Acceptance	8.3.42	-	
Tuning of warnings	Acceptance	8.3.43	-	
Stock status monitoring for optimization	Acceptance	8.3.44	-	

Table 4 – AHMS Test definition (Spare Management).

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

6. TEST AND VALIDATION PLAN UC8 - MATERIAL FLOW ANALYTICS AND SIMULATION (TRUMPF)

6.1. Background of the use case

The main objective of UC8 is summarized as a flexible production management of complex processes on the shop floor. A shop floor is the area of the production hall, where the machines are located. The main feature of UC8 is the realization of a digital twin of the shop floor. The digital twin (cyber component) is the digital representation that describes the shop floor (physical component). The digital twin can then be used together with simulation models and live data from the shop floor. The main goals are the reduction of efforts to set up a simulation model and to get live data from indoor localization systems in order to capture the live state of production.

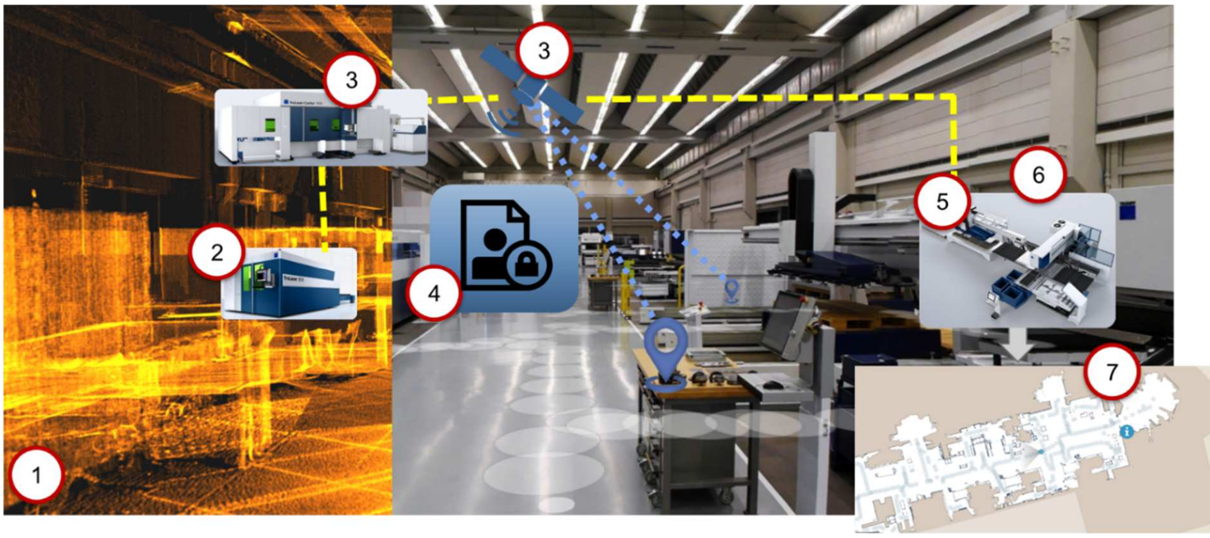


Figure 5 Overview over the major functions of UC8

The major functions of the CPS that is realized in UC8 are depicted in *Figure* ***5***: The pointcloud (1) that is generated by a 3D shopfloor scan, the enrichment of the model with semantics (2), the provision of an accurate material flow tracking system (3), the assessment w.r.t. ethical requirements (4), the automatic creation of a simulation model (5), the continuous adaption of the shop floor model based on this data (6) and finally the usage of the simulation results for real-time re-scheduling, re-routing and re-nesting (7). The test and validation of the overall use case will be demonstrated in the TRUMPF Customer Center in Ditzingen that is depicted in the background of *Figure* ***5***.

6.2. Strategy for test and validation

6.2.1. High level view of the strategy

Depending on the requirement type different approaches for testing and validation are carried out:

- Ethical requirements (ETH) 1-3 already include tests and validation. Therefore, they are out of scope of this document.
- Operational requirements (OPR) cannot be tested. In the following their level of fulfilment is given.
- For Functional (FNC), Integration (INT) and Performance (PFR) requirements tests and their respective success metrics are defined and carried out.
- To prove the function of the global use case a test environment is specified.

On a high level UC8 consists of 3 different components that have to be tested on their own and integrated to work with each other. The core component is the simulation model library that contains models of sheet metal manufacturing machines, automation units and storage systems which are developed in WP5. The project partner acs plus delivers the semantic enrichment module that detects and recognizes machines on the

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

shopfloor from 3D hall scans and 2D images. Finally, the simulation model is automatically built up from the position and machine type data that has been detected and recognized by the semantic enrichment module.

In order to fulfil the objectives of the use case it has to be validated that

- the simulation models of the distinct machines accurately model the real systems,
- a simulation model that consists of multiple machines and intralogistics units like forklifts or storage retrieval machines can be built up automatically,
- the semantic enrichment module accurately detects and recognizes the machines from 3D hall scans and 2D images,
- the whole pipeline from 3D hall scan to the simulation model of the shopfloor works seamlessly and that
- indoor localization data is analysed in real time and fed into the digital twin.

6.2.2. Tools and resources

The simulation models are tested on different levels of integration. The model units are tested in dedicated test cases. For each unit multiple test cases are setup when it is first released. A test case consists of input data and expected outputs that are compared to the test result. The test results are either analysed automatically to check for example whether a statistics entry is correct or manually to check for example whether the animation works properly. To assure that the messaging mechanisms between the different units and the material flow control work properly, integration test cases with multiple machines and automation units are performed. Each test case has a dashboard that is depicted in *Figure* ***6***. For each release all test cases have to be fulfilled successfully; this is depicted as a green dot in the figure.

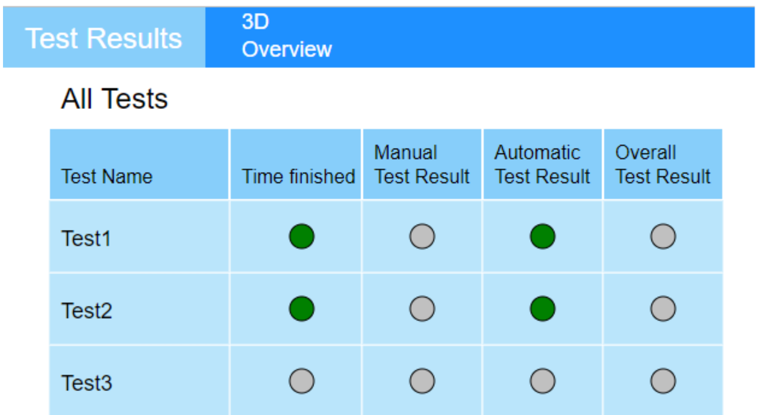


Figure 6 - Test Case Dashboard

These test cases currently only check for logical errors. At the moment we are lacking experimental data to validate the physical and temporal behaviour of the simulation models. To overcome this problem, we are planning to conduct real experiments and record the signals from the PLC to identify possible deviations.

Tools and resources to test the semantic enrichment module

The goal is to detect TRUMPF and 3rd party machines in a 3D shop floor scan. In order to achieve this, also 2D images from the hall scan are used. Thus, acs plus is continuously testing their technologies both for 2D and 3D data. See deliverable D3.3 for more details.

For the 2D technologies, acs plus created and is continuously updating a dataset of 2D images. Furthermore, a large effort is put into labelling the dataset. From this dataset, separate train and test datasets are extracted. Then, the performance of the technologies is measured on the separate test dataset, that was not used to train the models. These tests will be performed by software using common python libraries, such as pandas and scikit-learn.

The 3D technology requires a 3D point cloud of a TRUMPF hall scan, provided by a third party. It is planned to test the performance on the 3D-Scan using labels that annotate the type and position of machines in the hall scan. However, currently no such labels exist.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Tools and resources to test the UWB Infrastructure component

The UWB infrastructure component is based on the TRUMPF Track and Trace product. Therefore, it undergoes a regular testing and quality assurance process for each new product release. This includes many integration and component tests which are carried out in a SCRUM framework and documented in a ticket system (JIRA). Comprehensive integration and system tests are conducted based on JIRA XRAY test descriptions, test plans and test executions. Unit and component tests are partially automated utilizing Test Caffee and BrowserStack for UI tests and customized test suits for other components and E2E testing. For UC8 testing is focused on 1) the overall performance of the system (cf. document D3.3) and 2) on interfaces to the other demonstrator components using the cloud upload.

Tools and resources to test the overall use case

The TRUMPF customer centre in Ditzingen is chosen as testing environment for the overall use case. It offers a large variety of machines and automation units as well as an installed UWB tracking infrastructure. Labelled data is also available for the testing environment.

6.2.3. Tests definition

The tests that will be conducted are listed in the table below.

A reference to the use case requirements that the use case is intended to test can be found. It also shows the link to the detailed test definition that can be found in annex 8.4.

Test name	Test level	Test ref.	Req.ID	Comment / Reference
Semantic Enrichment Module Test	component	8.4.1	UC8-FNC-01 UC8-FNC-02	-
Interface Test	integration	8.4.3	UC8-INT-01	-
Ethical requirements validation	Does not apply	Does not apply	UC8-ETH-01 UC8-ETH-02 UC8-ETH-03	
Selection and test of hall scan provider	-	-	UC8-OPR-01 UC8-PFR-01	provider is selected only one single hall scan by this provider is available yet
UWB infrastructure test	component	8.4.2	UC8-OPR-02	-
Simulation model generation test	system	8.4.5	UC8-FNC-03 UC8-FNC-04	-
Simulation model performance test	component	8.4.6	UC8-PFR-02	-
Overall use case test	integration	8.4.7	UC8-INT-02	-

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Table 5 – Test definition.

7. TEST AND VALIDATION PLAN UC9 - MOBILE CPSS (WIKA)

7.1. Background of the use case

Collaborative Lifting is a use case provided by WIKA Mobile Control GmbH for this project. It deals with the use of at least two mobile machines, i.e. cranes, to lift a huge object that cannot be lifted using a single mobile crane.

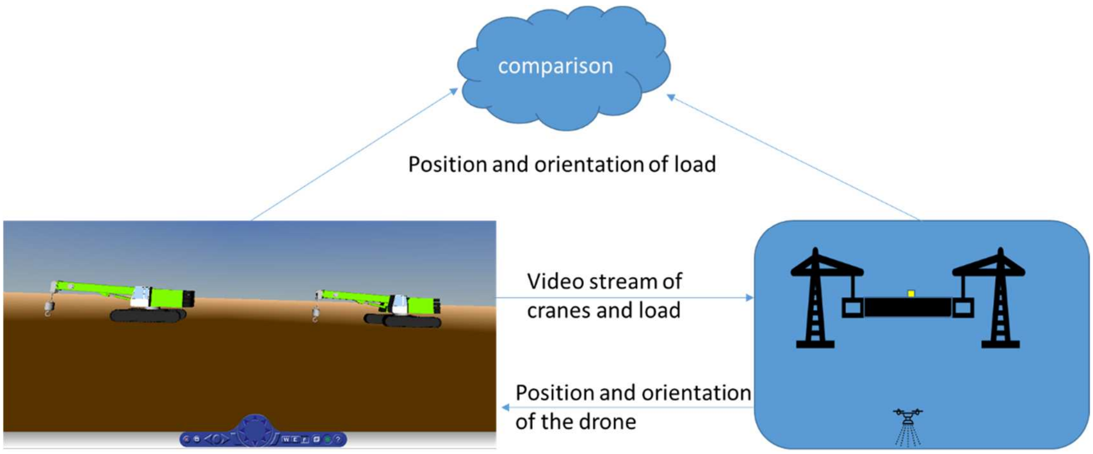
Nowadays, the planning of such a complex process is done either by classical methods for some cases (Pen & Paper) or using a planning and modelling software for others. Nevertheless, the execution of such a process still represents a challenge among the crane operators and fleet managers.

To accomplish a collaborative lifting process, it is mandatory that a lifting supervisor/ planer looks at the lifted object and at the machines and makes sure that the lifting is performed according to the plan. In many cases, the crane operators can have a restricted sight on the obstacles, humans and maybe other machines present on site, due to the size, volume and shape of the object lifted e.g., or due to complex movements that have to be performed. Thus, the lifting supervisor has to give instructions or hints to the crane operators via Walky-Talky or other means of communication to ensure a damage free lifting.

WIKA is proposing an innovative way to accomplish such complex task, relying on well-established technologies such as modelling, simulation, collaborative algorithms and new innovative technologies such as digital twins, AI-powered algorithms, real-time capable communication interfaces and cloud services. The integration and adaption of such technologies will make it possible to deliver the instructions for the collaborative lifting process on an HMI placed in the crane cabin and the lifting process will be supervised and monitored by a server (it can be local server on site or remote such as a cloud).

7.2. Strategy for test and validation

This use case will simulate a lift of an object with two cranes within a MATLAB Simulink program. The object shall pass a certain number of intermediate positions to reach the final destination. When an intermediate position is reached in the simulation, the position and orientation the lifted object is calculated. Then the Gazebo simulation will send the position and orientation of the drone. With that data MATLAB will produce a rendered video stream which will be transmitted to GAZEBO. From that video stream the drone will calculate the position and orientation of the load. There will be a process that compares the positions data of the object derived from the simulation and from a drone using a camera stream to calculate the position of the load. At each intermediate step the position is compared to the expected values.



All communication between MATLAB and GAZEBO are realized with ROS-topics.

7.2.1. High level view of the strategy

The prototype will consist of the MATLAB-simulation of the two cranes in which a video stream will be produced to be used by the drone to detect the position of the lifted object. Second there will be a Gazebo-simulation in which the drone is hovering. The two simulation will be communicating via the ROS api. This will certainly lead to a lag between the two simulations, which will be measured.

System part / Test Type	Coverage	Comment
MATLAB-simulation of cranes	This will be the system providing the sensor values and position data, which will be used in the Gazebo simulation.	This can be used to plan the lift in real world.
Gazebo-Simulation of the drone.	This simulation will be using the video data from the MATLAB-Simulation, to calculate the position of the load.	If this is accurate enough, a drone at the site can be used to detect the position of the load and compare it to the plan.
Compare process	This process will compare the calculated position of the load from MATLAB with the position from Gazebo.	Is needed to find the accuracy of the position detection via drone.
Measure delay between the two simulations	A method to measure the time delay has to be developed.	

Table 6 – Test strategy.

7.2.2. Tools and resources

The tools used are MATLAB and Gazebo. For the communication the ROS topics will be used.

7.2.3. Tests definition

The table below lists the test cases that are planned. A reference to the use case requirements that the use case is intended to test can be found. It also shows the link to the detailed test definition that can be found in annex 0.

Test name	Test level	Test ref.	Req.ID	
Simple Lift	system	8.5.1	UC9-FNC-09 UC9-FNC-10	
2 axis lift	system	8.5.2	UC9-FNC-09 UC9-FNC-10	
3 axis lift	system	8.5.3	UC9-FNC-09 UC9-FNC-10	

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

3 axis and rotation lift	system	8.5.4	UC9-FNC-09 UC9-FNC-10	

Table 7 – Test definition.

8. ANNEXES

8.1. UC4 Test case details [LEONARDO]

8.1.1. Vacuum positioning

Test Name	Vacuum positioning
Test Type	Component
Test purpose	Verify that the Cobot can reach the target position of the hole of the fuselage, taking into account the shape of the fuselage section
Test input	Target coordinates of the hole on the inner side of the fuselage section
Test description	<u>Test prerequisites</u>: reference system for the working area; program interface of the cobot controller; Cobot mounted on pedestal placed in front of the working area; vacuum hose attached to the cobot end effector. Enter in the program interface of the cobot the coordinates (x,y,z) of the target hole Observe the movement of Cobot and check the cobot arm trajectory. Check the landing position of the vacuum hose attached to the cobot end effector Repeat for other spatial coordinates positioned across various stringers on the fuselage
Expected output	Verify that the landing position of the vacuum hose meets the target hole. Verify the cobot arm trajectory avoids the stringers on the inner side of the fuselage section when moving between holes across one or more a stringer

8.1.2. Vacuum coverage of fuselage area

Test Name	Vacuum coverage of fuselage area
Test Type	Component
Test purpose	Verify that the Cobot can reach all the holes of the fuselage working area
Test input	Coordinates of at least four holes on the boundary of the working area
Test description	<u>Test prerequisites</u>: Cobot mounted on pedestal placed in front of the working area (in the middle of the working area); vacuum hose attached

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	to the cobot end effector Repeat test 8.1.1 for each selected hole on the boundary of the working area.
Expected output	Verify that the landing position of the vacuum hose meets the target hole.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.1.3. Drill localization

Test Name	Drill localization
Test Type	Component
Test purpose	Verify that the drill add-on communicates the right coordinates of the hole where the drill tip has been positioned
Test input	Coordinates of the hole
Test description	<u>Test prerequisites</u>: drill equipped with add-on; drill add-on connected to MQTT broker (WiFi); Print a tag with the coordinates of the hole Attach the tag over the hole Switch on the drill add-on and wait for a complete activation Approach the drill to the hole on the fuselage Press the operation button to activate the laser beam Direct the laser beam to the tag over the hole

Expected output	After successful reading of the tag, the display of the drill add-on shows the message that the operator must wait for the consensus to drill.
	On the MQTT broker/topic a packet is published with the coordinates of the hole.
	The payload of the packet has the following json format
	MQTT_TOPIC_PROXIMITY_LOCALIZATION =
	'proximity/UUID_DEVICE/localization'
	Proximity_Localization = {
	  “utime”: 1477671404,
	  “x”: 0.0,
	  “y”: 0.0,
	  “z”: 0.0
	}
	Verify that the coordinates x, y, z found in the packet are those entered in the tag.
	Verify the utime recorded in the packet is consistent with the time when the add-on read the tag.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.1.4. Drill close to the fuselage

Test Name	Drill close to the fuselage
Test Type	Integration
Test purpose	Verify that the drill add-on communicates to the gateway the proximity of the drill tip to the fuselage to the gateway when the drill tip is close to the fuselage.
Test input	
Test description	Test prerequisites: drill equipped with add-on; gateway connected to the network (cable), drill add-on connected to the network (WiFi); Switch on the drill add-on and wait for a complete activation Approach the drill to the hole on the fuselage Press the operation button to activate the laser beam and read the tag near the hole. Move the drill tip at different distances from that hole on the fuselage and then rest the drill tip on the hole. Then move away the drill tip from the fuselage.

Expected output	On the MQTT broker/topic a stream of MQTT packets is published showing the distance of the tip to the fuselage.
	The payload of the packet has the following json format
	MQTT_TOPIC_PROXIMITY_DATA = 'proximity/UUID_DEVICE/data'
	Proximity_Data = {
	  “utime”: 1477671404,
	  “pitch”: 0.0,
	  “roll”: 0.0,
	  “distance”: 0.0
	}
	Verify that the distance found in the packets series shows the change of the distance according to the movements made.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.1.5. Operator ready to drill at a position

Test Name	Operator ready to drill at a position
Test Type	Integration
Test purpose	Verify that the drill add-on sends to the gateway the messages indicating the operator is ready to drill at the detected position
Test input	
Test description	<u>Test prerequisites</u>: reference system for the working area; tag positioned over the hole; drill equipped with add-on; gateway connected to the network (cable), drill add-on connected to the network (WiFi) Switch on the drill add-on and wait for a complete activation Approach the drill to the hole on the fuselage Press the operation button to activate the laser beam and read the tag above the hole. Wait for message of successful reading on the display of the drill add-on

Expected output	On the MQTT broker/topic on the gateway a packet is published with the coordinates of the hole.
	The payload of the packet has the following json format
	MQTT_TOPIC_PROXIMITY_LOCALIZATION = 'proximity/UUID_DEVICE/localization'
	Proximity_Localization = {
	  “utime' : 1477671404,
	  “x”: 0.0,
	  “y”: 0.0,
	  “z”: 0.0
	}
	Verify that the coordinates x, y, z found in the packet are those found in the tag.
	Check on the MQTT broker/topic a packet is published with this json payload:
	MQTT_TOPIC_PROXIMITY_ACTION = 'proximity/UUID_DEVICE/action'
	Proximity_Action = {
	  “utime”: 1477671404,
	  “action: 'start_hole'
	}

8.1.6. Vacuum positioning command

Test Name	Vacuum positioning command
Test Type	Integration

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Test purpose	Verify that the gateway communicates to the vacuum controller the target coordinates of the hole
Test input	Coordinates of the target hole
Test description	Test prerequisites: gateway and vacuum controller connected to the network (cable); stub on the gateway simulating the drill add-on; Cobot mounted on pedestal placed in front of the working area; vacuum hose attached to the cobot end effector. Enter the coordinates of the target hole on the stub Activate the stub
Expected output	Verify that the cobot moves the vacuum hose to the target hole.

8.1.7. Vacuum positioning feedback

Test Name	Vacuum positioning feedback
Test Type	Integration
Test purpose	Verify that the vacuum controller communicates to the gateway that the cobot has reached the target position
Test input	
Test description	Test prerequisites: gateway and vacuum controller connected to the network (cable); stub on the gateway simulating the drill add-on; Cobot mounted on pedestal placed in front of the working area; vacuum hose attached to the cobot end effector. Execute test 8.1.6.
Expected output	Verify that when the cobot has reached the final position a socket message on the gateway is received originating from the cobot with the feedback of the vacuum controller. The payload of the message has the following format (2,x,y,z)\n Where x,y,z are the coordinates of the target hole.

8.1.8. Consensus to drill

Test Name	Consensus to drill
Test Type	Integration
Test purpose	Verify that the gateway communicates to the drill add-on the consensus to drill when the cobot has reached the target position
Test input	
Test description	Test prerequisites: gateway and vacuum controller connected to the network (cable); drill equipped with add-on; Cobot mounted on pedestal

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	placed in front of the working area; vacuum hose attached to the cobot end effector. Approach the drill to the hole on the fuselage Press the operation button and read the tag above the hole. Wait for a consensus message on the drill that the operator is allowed to drill
Expected output	After the vacuum hose attached to the cobot has reached the target position the drill add-on shows on the display the text message that the operator is allowed to drill.

8.1.9. Vacuum positioning after the operator is ready to drill

Test Name	Vacuum positioning after the operator is ready to drill
Test Type	System
Test purpose	Verify that the cobot does not move if the consensus is not provided and moves to the target hole after the operator is ready to drill at that hole
Test input	
Test description	<u>Test prerequisites</u>: tag positioned over the hole; drill equipped with add-on; vacuum controller and cobot; gateway and vacuum controller connected to the network (cable), drill add-on connected to the network (WiFi); fuselage Switch on the drill add-on and wait for a complete activation Switch on the vacuum controller and cobot and wait for a complete activation Approach the drill to the position on the fuselage where the hole is made Press the operation button to activate the laser Wait some time before directing the laser beam on the tag over the hole; then scan the tag over the hole Wait for message of successful reading on the display of the drill add-on
Expected output	The cobot starts moving towards the target hole after the message of successful reading is displayed; the cobot does not move before that message.

8.1.10. Positive consensus to drill

Test Name	Positive consensus to drill
Test Type	System
Test purpose	Verify that after the cobot has reached the target hole, the operator receives the consensus to drill.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Test input	
Test description	<u>Test prerequisites</u>: tag positioned over the hole; drill equipped with add-on; vacuum controller and cobot; gateway and vacuum controller connected to the network (cable), drill add-on connected to the network (WiFi); fuselage. Switch on the drill add-on and wait for a complete activation Switch on the vacuum controller and cobot and wait for a complete activation Approach the drill to the hole on the fuselage Press the operation button to activate the laser Direct the laser beam on the tag over the hole until the message of successful reading is displayed on the drill add-on. Wait for a consensus message on the drill add-on display saying that the operator is allowed to drill
Expected output	The cobot moves to the target hole. After the cobot has reached the target hole the drill add-on **shows** the text message that the operator is allowed to drill.

8.1.11. Negative consensus to drill

Test Name	Negative consensus to drill
Test Type	System
Test purpose	Verify that if the cobot cannot reach the target hole the operator does not receive the consensus to drill and is informed that he cannot drill
Test input	
Test description	<u>Test prerequisites</u>: tag positioned over the hole; drill equipped with add-on; vacuum controller and cobot; gateway and vacuum controller connected to the network (cable), drill add-on connected to the network (WiFi); fuselage; obstacle on the path of the cobot trajectory to the target hole Switch on the drill add-on and wait for a complete activation Switch on the vacuum controller and cobot and wait for a complete activation Approach the drill to the hole on the fuselage Direct the laser beam on the tag over the hole until the message of successful reading is displayed on the drill add-on. Wait for a consensus message on the drill add-on display saying that the operator is allowed to drill.
Expected output	The cobot stops due to the obstacle on the path to the target hole. The drill add-on **does not** show on the display the text message that the operator is allowed to drill but shows a message that drilling is denied.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.1.12. Drill tip wear estimation (good tip)

Test Name	Drill tip wear estimation (good tip)
Test Type	Component
Test purpose	Verify that the system returns that drill tip is still suited for drilling when the tip that is subject to the wear control cycle has an acceptable remaining useful life.
Test input	A brand new tip and other drill tips that show different wear levels but all deemed to have an acceptable remaining useful life.
Test description	Test prerequisites: tool wear module, drill tips Switch on the tool wear module Mount a drill tip on the support of the tool wear module Adjust the drill tip on the support to ensure centering and alignment with the camera of the wear module. Start the wear control cycle and wait for the completion of the cycle Check the outcome of the control on the display of the tool wear module. Repeat the test for each drill in the input set.
Expected output	For brand new tip and for each drill tip in the input set the outcome of the wear control is that the drill tip can still be used.

8.1.13. Drill tip wear estimation (worn out tips)

Test Name	Drill tip wear estimation (worn out tips)
Test Type	Component
Test purpose	Verify that the system returns that drill tip is still no more suited for drilling when the tip that is subject to the wear control cycle does not have an acceptable remaining useful life.
Test input	A worn out tip and other drill tips that show different wear levels but all deemed to have a unacceptable remaining useful life and should not be used any more for drilling.
Test description	Test prerequisites: tool wear module, drill tips Switch on the tool wear module Mount a drill tip on the support of the tool wear module Adjust the drill tip un the support to ensure centering and alignment with the camera of the wear module. Start the wear control cycle and wait for the completion of the cycle Check the outcome of the control on the display of the tool wear module. Repeat the test for each drill in the input set.
Expected output	For the worn out tip and for each drill tip in the input set the outcome of the wear control is that the drill tip is not good for drilling.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.1.14. Dynamic obstacle perception

Test Name	Dynamic obstacle perception
Test Type	Component
Test purpose	Verify that when an object intercepts the cobot movement, the cobot stops
Test input	
Test description	<u>Test prerequisites</u>: cobot equipped with cobot controller; Switch on the cobot controller and the cobot until a complete activation Enter in the cobot controller the target coordinates; Start the movement of the cobot; While the cobot is moving put an object on the trajectory of the cobot arm
Expected output	The cobot detects the object and stops without hitting the object

8.1.15. Static obstacle perception

Test Name	Static obstacle perception
Test Type	Component
Test purpose	Verify that when the cobot finds an object on his trajectory the cobot stops
Test input	
Test description	<u>Test prerequisites</u>: cobot equipped with cobot controller; an object is placed on the trajectory of the cobot arm to the target coordinates Switch on the cobot controller and the cobot until a complete activation Enter in the cobot controller the target coordinates; Start the movement of the cobot;
Expected output	The cobot stops when it detects the object on its trajectory and stops without hitting the object

8.1.16. Vacuum Positioning Time

Test Name	Vacuum Positioning Time
Test Type	System / performance
Test purpose	Verify that the consensus to drill is communicated to the operator within six seconds since he gave his consensus to drill
Test input	
Test description	<u>Test prerequisites</u>: reference system for the working area; tag positioned over the hole; drill equipped with add-on; cobot equipped with controller; gateway

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	and cobot controller connected to the network (cable), drill add-on connected to the network (WiFi); fuselage Switch on the drill add-on and wait for a complete activation Switch on the vacuum controller and cobot and wait for a complete activation Approach the drill to the hole on the fuselage Press the operation button and read the tag above the target hole. Check for message of successful reading on the display of the drill add-on Wait for a consensus message on the drill that the operator is allowed to drill.
Expected output	The consensus to drill is displayed before six seconds have passed after the display of successful reading of the tag.

8.1.17. Drilling process cycle

Test Name	Drilling process cycle
Test Type	System/Acceptance
Test purpose	Verify that the system supports all the steps of the drilling process and that they are displayed on the process monitoring interface while they occur.
Test input	
Test description	Test prerequisites: reference system for the working area; tag positioned over the hole; drill equipped with add-on; cobot equipped with controller; gateway and cobot controller connected to the network (cable), drill add-on connected to the network (WiFi); fuselage Switch on the drill add-on and wait for a complete activation Switch on the cobot and cobot controller and wait for a complete activation Approach the drill to the hole on the fuselage Press the operation button to activate the laser Direct the laser to the tag over the hole Check for message of successful reading on the display of the drill add-on Check for message of consensus to drill on the display of the drill add-on Execute the drilling Press the operation button to communicate the drilling is over Move the drill away from the fuselage
Expected output	The following events are displayed on the process monitoring interface (HMI) and logged on the enterprise data analysis platform: • drill is moving • drill positioned and Operator is Ready to drill • vacuum is moving • vacuum positioned • drilling allowed • drilling is over

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2. UC5 Test case details [LEONARDO]

8.2.1. Environment parameters measurement

Test Name	Environment parameters measurement
Test Type	Component
Test purpose	Verify that the Distributed measurement system for the working area parameters (sensor node) produces measurement data of the work environment
Test input	Environment
Test description	Test prerequisites: NTP service, MQTT broker, DMS connected to the network (WiFi), instruments to measure the temperature, pressure and humidity of the environment Configure on the Distributed measurement system the remote NTP service and MQTT broker/topic that it needs to reach Activate the NTP service and MQTT broker Deploy the Distributed measurement system in the tested environment Using offline instruments take measures of the temperature, pressure and humidity of the environment at the position where the Distributed measurement system is deployed Turn the Distributed measurement system on Check MQTT packets published on the MQTT broker/topic configured.

Expected output	On the MQTT broker/topic a series of MQTT/packet is published at the frequency of 1 packet/5s, expected for the work environment data flow
	The payload of each packet has the following json format:
	MQTT_TOPIC_ENVIRONMENTAL = 'enviromental/UUID_DEVICE/data'
	Environmental_Data = {
	“utime”: 1477671404,
	“temperature”: 30.0,
	“humidity”: 99.0,
	“pressure”: 128.0
	}
	Temperature, pressure and humidity recorded in the packet are consistent with the measures taken with the offline instruments.
	The utime recorded in the packets is consistent with the time when the Distributed measurement system was turned on.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.2. Collection on the field of environmental parameters

Test Name	Collection on the field of environmental parameters
Test Type	Integration
Test purpose	Verify that the Distributed Measurement System (DMS) for the working area parameters communicates the measurement of the environment parameters to the gateway
Test input	Environment
Test description	<u>Test prerequisites</u>: NTP service, gateway connected to the network (cable), DMS connected to the network (WiFi), Configure on the Distributed Measurement System the remote NTP service and MQTT broker/topic that it needs to reach Activate the NTP service and gateway Deploy the Distributed measurement system in the testing environment Turn the Distributed Measurement System on

Expected output	On the MQTT broker/topic on the gateway a series of MQTT/packet is published at the frequency of 1 packet/5s
	The payload of each packet has the following json format:
	MQTT_TOPIC_ENVIRONMENTAL = 'enviromental/UUID_DEVICE/data'
	Environmental_Data = {
	“utime”: 1477671404,
	“temperature”: 30.0,
	“humidity”: 99.0,
	“pressure”: 128.0
	}
	The utime recorded in the packets is consistent with the time when the Distributed Measurement System was turned on.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.3. Worked part vibration measurement

Test Name	Worked part vibration measurement
Test Type	Component
Test purpose	Verify that the Distributed Measurement Systems for the work part parameters produce the window vibration measurement data
Test input	Vibrations induced artificially
Test description	Test prerequisites: DMS vibration sensor connected to the network (WiFi), NTP service, MQTT broker, vibrations generator Secure the Distributed Measurement System (DMS) on a workbench close to the vibration generator Configure on the Distributed Measurement System the remote NTP service and MQTT broker/topic that it needs to reach Activate the NTP service and MQTT broker Turn the Distributed measurement system on Activate the vibration generator Check MQTT packets published on the MQTT broker/topic configured.
Expected output	On the MQTT broker/topic a series of MQTT/packets is published at the frequency of 100Hz. The payload of each packet has a binary format containing raw data of the accelerations measured on the 3 axis. Accelerations recorded in the packet are consistent with range of frequencies produced by the generator. The utime recorded in the packets is consistent with the time when the Distributed measurement system was turned on.

8.2.4. Collection on the field of the worked part vibration measure

Test Name	Collection on the field of the worked part vibration measure
Test Type	Integration
Test purpose	Verify that the Distributed measurement systems for the work part parameters communicate the window vibration measurement to the gateway
Test input	Vibrations induced artificially
Test description	Test prerequisites: DMS vibration sensor connected to the network (WiFi), NTP service, gateway connected to the network (cable), vibrations generator Secure the Distributed Measurement System (DMS) on a workbench close to the vibration generator Configure on the Distributed Measurement System the remote NTP service and MQTT broker/topic that it needs to reach

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	Activate the NTP service and MQTT broker Turn the Distributed measurement system on Activate the vibration generator Check MQTT packets published on the MQTT broker/topic on the gateway Repeat the test adding other DMS nodes up to the number of twelve (nr of windows on one side of the fuselage section).
Expected output	On the MQTT broker/topic a series of MQTT/packets on the gateway is published at the frequency of 100Hz for each DMS node activated. The payload of each packet has the binary format as in 8.2.3. The data streams are buffered and saved on the gateway.

8.2.5. Trimming head vibration measurement

Test Name	Trimming head vibration measurement
Test Type	Component
Test purpose	Verify that the Distributed Measurement System for the trimming head parameters produce the measurement data of the vibration of the machine head
Test input	Vibrations induced artificially
Test description	<u>Test prerequisites</u>: DMS vibration sensor connected to the network (WiFi), NTP service, MQTT broker, vibrations generator Secure the Distributed Measurement System (DMS) on a workbench close to the vibration generator Configure on the Distributed Measurement System the remote NTP service and MQTT broker/topic that it needs to reach Activate the NTP service and MQTT broker Turn the Distributed Measurement System on Activate the vibration generator Check MQTT packets published on the MQTT broker/topic configured.
Expected output	On the MQTT broker/topic a series of MQTT/packets is published at the frequency of 100Hz. The payload of each packet has a binary format containing raw data of the accelerations measured on the 3 axis. Accelerations recorded in the packet are consistent with range of frequencies produced by the generator. Vibrations recorded in the packet are consistent with range of frequencies produced by the generator. The utime recorded in the packets is consistent with the time when the Distributed measurement system was turned on.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.6. Collection on the field of the trimming head vibration measure

Test Name	Collection on the field of the trimming head vibration measure
Test Type	Integration
Test purpose	Verify that the Distributed Measurement System (DMS) for the trimming head parameters communicates the measurement of the vibration of the machine head to the gateway
Test input	Vibrations induced artificially
Test description	<u>Test prerequisites</u>: DMS vibration sensor connected to the network (WiFi), NTP service, gateway connected to the network (cable), vibrations generator (such as an electrical engine) Secure the Distributed Measurement System (DMS) on a workbench close to the vibration generator Configure on the Distributed Measurement System the remote NTP service and MQTT broker/topic that it needs to reach Activate the NTP service and gateway Turn the Distributed Measurement System on Activate the vibration generator
Expected output	On the MQTT broker/topic a series of MQTT/packets on the gateway is published at the frequency of 100Hz for each DMS node activated. The payload of each packet has the binary format as in 8.2.5. The data streams are buffered and saved on the gateway.

8.2.7. Trimming tool tip vibration measurement

Test Name	Trimming tool tip vibration measurement
Test Type	Component
Test purpose	Verify that the trimming parameters acquisition chain outputs the vibration measurement data of the machine tip.
Test input	Vibrations induced artificially
Test description	<u>Test prerequisites</u>: NTP service, vibrations generator, sensorized mandrel Schunk iTENDO complete with transceiver and console connected to the network (cable) Secure the tip mounted on the mandrel to the vibration generator Configure on the Mandrel System (mandrel, transceiver and console) the remote NTP service and the sampling frequency (10Hz) Activate the NTP service Turn the mandrel system on Activate the vibration generator Check the registered vibration data on the console

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Expected output	Every 1/10 seconds a record is displayed on the console screen. The record shows a timestamp and an IFT value for the vibration. IFT is a numerical value a measurement used by iTENDO to show the measured vibration as a numeric value on a defined intensity scale.

8.2.8. Collection on the field of the trimming tool tip vibration measure

Test Name	Collection on the field of the trimming tool tip vibration measure
Test Type	Integration
Test purpose	Verify that the trimming parameters acquisition chain communicates to the gateway the measurement of the vibration of the machine tip
Test input	Vibrations induced artificially
Test description	<u>Test prerequisites</u>: NTP service, gateway connected to the network (cable), vibrations generator (such as an electrical engine), sensorized mandrel Schunk iTENDO complete with transceiver and console connected to the network (cable). Secure the tip mounted on the mandrel to the vibration generator Configure on the Mandrel System (mandrel, transceiver and console) the remote NTP service and the sampling frequency (100Hz); configure the MQTT broker/topic that it needs to reach Activate the NTP service Turn the mandrel system on Activate the vibration generator Check the gateway for incoming messages

Expected output	A series of MQTT/packest is published at the frequency of 100Hz on the MQTT broker/topic configured on the gateway.
	The payload of each packet contains the IFT value and raw data samples of the acceleration measured on one axe. The format of the packet is the following:
	MQTT_TOPIC_PCSIDEM = 'jobs/UUID_DEVICE/data
	{
	"utime": 1629809393266,
	# Shunk data
	"value": 30.0, # Vibration Index
	"v": [# Row Vibration array
	32123,
	32173,
	32843,
	...
	],
	"f": 100.75, #Row Vibration Sampling
	}
	The data streams are buffered and saved on the gateway.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.9. Trimming air flow measurement

Test Name	Trimming air flow measurement
Test Type	Component
Test purpose	Verify that the Distributed Measurement Systems (DMS) for the trimming hood parameters produces the measurement data of the flow and temperature of the air in the hood of the machine
Test input	Air flow
Test description	Test prerequisites: DMS air flow sensor connected to the network (WiFi), NTP service, MQTT broker, air blower and hose Insert the Distributed Measurement System (DMS) on the hose Configure on the Distributed Measurement System the remote NTP service and MQTT broker/topic that it needs to reach Activate the NTP service and MQTT broker Turn the Distributed Measurement System on Activate the air blower and blow the air into the hose Check the MQTT packets published on the MQTT broker/topic.

Expected output	On the MQTT broker/topic a series of MQTT/packets is published every 2.5 seconds.
	Each packet contains the temperature and flow measurements of the airflow. The packet has the following json format:
	MQTT_TOPIC_TRIMMER_DATA = 'trimmer/UUID_DEVICE/vacuum'
	TrimmerVacuum_Data = {
	  "utime": 1477671404,
	  "temperature": 30.0,
	  "flux": 99.0,
	}
	Temperature and flux values recorded in the packets are consistent with those of the airflow produced by the air blower.
	The utime recorded in the packets is consistent with the time when the Distributed measurement system was turned on.

8.2.10. Collection on the field of the Trimming air flow measure

Test Name	Collection on the field of the Trimming air flow measure
Test Type	Integration
Test purpose	Verify that the distributed measurement systems for the trimming hood parameters communicate the measurement data of the flow and temperature of the air in the hood of the machine to the gateway

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Test input	Air flow at a predefined flow and temperature
Test description	Test prerequisites: DMS air flow sensor connected to the network (WiFi), NTP service, gateway connected to the network (cable), air blower Secure the Distributed Measurement System (DMS) on a workbench close to the air blower Configure on the Distributed Measurement System the remote NTP service and MQTT broker/topic that it needs to reach Activate the NTP service and MQTT broker Turn the Distributed Measurement System on Activate the air blower Check MQTT packets published on the MQTT broker/topic configured.
Expected output	On the MQTT broker/topic a series of MQTT/packets on the gateway is published at the frequency of 0,4Hz. The payload of each packet has the binary format as in 8.2.10. The data streams are buffered and saved on the gateway.

8.2.11. Trimming machine work parameters acquisition

Test Name	Trimming machine work parameters acquisition
Test Type	Component
Test purpose	Verify that the trimming parameters acquisition chain obtains from the machine the measurement data of the machining parameters of the machine
Test input	
Test description	Test prerequisites: NTP service, MQTT broker, computing station running the data collection program of trimming machine process parameters, trimming machine and numeric control program Activate the NTP service and MQTT broker Turn data collection program of trimming machine process parameters Execute the numeric control program on the trimming machine for the trimming of a window While the trimming process occurs reduce the forward speed at 50% (trimming operator override command) and record the time when the command is entered. Check MQTT packets published on the MQTT broker/topic configured.
Expected output	On the MQTT broker/topic a stream of MQTT/packets is published every 0.1 seconds. Each packet contains a measure of various process parameters that the data collection program obtains form the trimming machine via OPC-UA. The payload of each packet has the following json format: MQTT_TOPIC_PCSIDEM = 'jobs/UUID_DEVICE/data'

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	{
	"utime": 1629809393266,
	#Jobs Data
	"ss": 330.0, # Spindle speed of the trimming tool (rpm)
	"fs": 330.0, # Feed speed of the trimming tool (cm/sec)
	"rul": "330", # Remaining Useful Life of the tool (minutes)
	"tid": "6", # Tool id (unique identification number of the tool)
	"tt": "0", # Tool type (0 = roughing, 1 = finishing)
	"wid": 14.0, # Window id (unique identification number of the part that is worked)
	"ppb": "3645", # Part Program Block
	"pps": 9 # Part Program Sequence
	}
	Trimming parameters recorded in the packet match the part program executed by the numeric control of the trimming machine. The feed speed samples show the 50% reduction at the time when the trimming operator entered the override command.
	The utime recorded in the packets is consistent with the interval when trimming process was executed.

8.2.12. Collection of Trimming machine work parameters

Test Name	Collection of Trimming machine work parameters
Test Type	Integration
Test purpose	Verify that the trimming parameters acquisition chain communicates to the gateway on the edge the measurement data of the machining parameters of the machine
Test input	
Test description	<u>Test prerequisites</u>: NTP service, gateway connected to the network (cable), computing station running the data collection program of trimming machine process parameters connected to the network (cable), stub that simulates the execution of the trimming process by the trimming machine Activate the NTP service and the gateway Turn on data collection program of trimming machine process parameters Turn on the stub that simulates the interface to obtain the process parameters form the trimming machine Check MQTT packets published on the MQTT broker/topic configured on the gateway
Expected output	On the MQTT broker/topic a series of MQTT/packets on the gateway is published at the frequency of 10Hz. The payload of each packet has the binary format as in 8.2.11. The data streams are buffered and saved on the gateway.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.13. Communication of collected measures to the enterprise data analysis platform

Test Name	Communication of collected measures to the enterprise data analysis platform
Test Type	Integration
Test purpose	Verify that the gateway communicates to the enterprise data analysis platform the measurement data acquired from the field
Test input	
Test description	<u>Test prerequisites</u>: • NTP service, • gateway connected to the network (cable), • data collection program of trimming machine process parameters on separate computing station connected to the network (cable), • stub that simulates the execution of the trimming process by the trimming machine, • Data Measurement System nodes (window vibration, head vibration, environment, air flow) connected to the network (WiFi), • sensorized mandrel Schunk iTENDO complete with transceiver and console connected to the network (WiFi), • enterprise data analysis platform • air blower, vibration generator, workbench Activate the NTP service and the gateway Secure the DMSs and the mandrel on a workbench Activate the DMSs and the mandrel Activate the trimming machine stub Turn the data collection program of trimming machine process parameters on Activate the air blower close to the DMS air flow sensor Activate the vibration generator close to the DMS vibration sensors Access the data analysis platform to check that data are received from the gateway and stored.
Expected output	On the data store of the data analysis platform several files are found, one for each data measurement system deployed. Each file contains the data streams produced by the various DMS nodes: • Environment data (temperature, pressure, humidity) • Windows vibration (acceleration on the 3 axis) • Trimming head vibration (acceleration on the 3 axis) • Air flow (flux and temperature) • Mandrel vibration (IFT value, acceleration on 1 axis) and Trimming machine parameters

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.14. Data staging of measures collected from the field

Test Name	Data staging of measures collected from the field
Test Type	Component
Test purpose	Verify that the enterprise data analysis platform is able to organize the data collected from the field and allows the user to browse those data
Test input	Files received from the field containing data streams produced by the various DMS nodes
Test description	Test prerequisites: • enterprise data analysis platform with data staging module Load the files on the data store of the data analysis platform Access the data analysis platform and go to the section to view data Set an observation interval and check the data that are displayed
Expected output	The system displays a summary page where in the different columns of a single table the user can see how the values of the measured parameters recorded the various data streams change during the observation interval. The user can also select one or more parameters that are plotted on the same chart. The system also presents a detail page for each data stream where the samples of measures in that stream on the selected observation interval are presented in a table form and on a chart i.e. one page for: • Environment data (temperature, pressure, humidity) • Windows vibration (acceleration on the 3 axis) • Trimming head vibration (acceleration on the 3 axis) • Air flow (flux and temperature) The user can browse the data setting the time interval or filtering the range of values of interest.

8.2.15. Data collection of relevant trimming process parameters from the field

Test Name	Data collection of relevant trimming process parameters from the field
Test Type	System / Acceptance
Test purpose	Verify that the data of the relevant parameters measured while the trimming is carried out on the plant are collected on the field and transmitted to the central data analysis platform where they are stored and organized and can be viewed offline.
Test input	
Test description	Test prerequisites: • Real environment with the trimming machine working on fuselage section • NTP service,

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	• gateway connected to the network (cable), • data collection program of trimming machine process parameters on separate computing station connected to the network (cable), • DMSs (window vibration, head vibration, environment, air flow) connected to the network (WiFi), • sensorized mandrel Schunk iTENDO complete with transceiver and console connected to the network (WiFi), • enterprise data analysis platform with data staging module Set the DMSs vibration sensors above the fuselage windows Set the DMS for the head vibration on the head of the trimming machine Set the DMS for the work environment on the cage of the trimming machine Insert the DMS of the air flow on the out take pipe from the hood of the trimming head Mount the sensorized mandrel Schunk iTENDO on the trimming machine Activate the NTP service and the gateway Activate the data collection program of trimming machine process parameters Activate all the DMSs Start the work program for windows trimming on the trimming machine Access the data analysis platform and check that data are received and stored. Go to the section to view data and check the data that have been received.
Expected output	On the data store of the data analysis platform several files are found, one for each data measurement system deployed. Each file contains the data streams produced by the various DMS nodes (see 8.2.13). The table on the summary page with different columns for the various collected parameters shows the values of the parameters measured on the filed during the trimming process interval (see 8.2.14).

8.2.16. Data Loading of quality inspection data

Test Name	Data Loading of quality inspection data
Test Type	Component
Test purpose	Verify that the enterprise data analysis platform is able to load from file and archive the data of the defects found on the processes
Test input	Data files in excel format where defects found during the final quality inspection after the window trimming on a fuselage section are tracked. Files received from the field containing data streams produced by the various DMS nodes during the windows trimming on that section.
Test description	Test prerequisites: • enterprise data analysis platform with data staging module Load on the data store of the data analysis platform the files with the data streams collected on the field;

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	Put the data files where defects are recorded in a dedicated input directory Launch the upload from that directory Go to the section to view the data Check the data defects that have been loaded Check that defects data are merged with the data set of the trimming parameters collected during the trimming process.
Expected output	The system provides a dedicated page where the defects imported from the file can be viewed. For each reported defect the following information is displayed • window nr • start position • end position • depth • defect gravity In the summary page where the user can see the table showing how the values of the parameters measured during the trimming process have changed, the records concerning the interval when the defect occurred are marked and highlighted.

8.2.17. Data analisys

Test Name	Data analysis
Test Type	Component/System/Acceptance
Test purpose	Verify that the enterprise data analysis platform allows the user to analyse the correlation between parameters collected during the trimming process
Test input	Files received from the field containing data streams produced by the various DMS nodes
Test description	Test prerequisites: • enterprise data analysis platform with data staging module and data analysis module Load the files on the data store of the data analysis platform Access the data analysis platform and go to the section to analyse the data Set an analysis interval of analysis within the period when the trimming occurred and data have been collected for trimming parameters Choose a couple of parameters of interest
Expected output	The system shows a correlation matrix of the chosen parameters over the selected interval of time.

8.2.18. Prediction model validation

Test Name	Prediction model validation
Test Type	System/Acceptance

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Test purpose	Verify the trained model provides the expected prediction by applying it on previously collected data where the production output quality is known
Test input	Files received from the field containing data streams produced by the various DMS nodes during the windows trimming on fuselage sections (see 8.2.14). Data files in excel format where defects found during the final quality inspection after the window trimming on those sections are tracked.
Test description	Test prerequisites: model trained with part of the collected data (80%) to provide the expected prediction with precision and recall indexes >80%; azure container where the model is deployed; stub to read the data streams from file and feed the model with it. - Execute the prediction model - Launch the stub to feed the model with a part of the data set (20%) that was not used before to train the model.
Expected output	Verify that when processing the input data set the prediction model raises alerts corresponding to the positions where defects on the windows trimming output have been reported, with precision and recall indexes >80%

8.2.19. HMI display

Test Name	HMI display
Test Type	Component
Test purpose	Verify that the HMI shows the process data and the prediction alerts produced by the model
Test input	Files received from the field containing data streams produced by the various DMS nodes during the windows trimming on fuselage sections (see 8.2.14). File with alert events produced by the production model
Test description	Test prerequisites: data analysis platform, operator HMI application, stubs to feed the operator HMI with the data from the input files; web browser - Start the data analysis platform and the operator HMI application - Access the operator HMI with the browser - Launch the stubs - Check the information displayed on the browser
Expected output	The HMI shows how the trimming process variables change during the window trimming and highlights if any values are getting close or exceed a predefined range. Alerts are raised on the HMI when alert events occur.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.20. Communication to the operator HMI of collected measures and prediction output

Test Name	Communication to the operator HMI of the collected measures and prediction output
Test Type	Integration
Test purpose	Verify that while the trimming process occurs the gateway is able to communicate to the HMI the data about process variables along with the prediction alerts produced by the prediction model running on the gateway
Test input	
Test description	<u>Test prerequisites</u>: • NTP service, • gateway connected to the network (cable), • data collection program of trimming machine process parameters on separate computing station connected to the network (cable), • stub that simulates the execution of the trimming process by the trimming machine, • Data Measurement System nodes (window vibration, head vibration, environment, air flow) connected to the network (WiFi), • sensorized mandrel Schunk iTENDO complete with transceiver and console connected to the network (WiFi), • enterprise data analysis platform • HMI application • Prediction model azure application deployed on the gateway (azure container) • air blower, vibration generator, workbench • web browser Activate the NTP service and the gateway Start the prediction model application on the gateway Secure the DMSs and the mandrel on a workbench Start the data analysis platform and the operator HMI application Activate the DMSs and the mandrel Activate the trimming machine stub Turn the data collection program of trimming machine process parameters on Activate the air blower close to the DMS air flow sensor Activate the vibration generator close to the DMS vibration sensors Access the operator HMI with a browser and check the information displayed on the browser
Expected output	The HMI shows the values of the trimming process variables measured by the DMS and the alerts, if any, raised by the prediction model elaborating those variables.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.21. Trimming process monitoring

Test Name	Trimming process monitoring
Test Type	System/Acceptance
Test purpose	Verify that as plant processing takes place, the operator HMI displays the data of the relevant process variables measured on the field and shows any alerts produced by the prediction model
Test input	-
Test description	<u>Test prerequisites</u>: • Real environment with the trimming machine working on fuselage section • NTP service, • gateway connected to the network (cable), • prediction model deployed on the gateway (as containerized Azure application) • data collection program of trimming machine process parameters on separate computing station connected to the network (cable) , • DMSs (window vibration, head vibration, environment, air flow) connected to the network (WiFi), • sensorized mandrel Schunk iTENDO complete with transceiver and console connected to the network (WiFi), • enterprise data analysis platform • HMI application • Web browser Set the DMSs vibration sensors above the fuselage windows Set the DMS for the head vibration on the head of the trimming machine Set the DMS for the work environment on the cage of the trimming machine Insert the DMS of the air flow on the out take pipe from the hood of the trimming head Mount the sensorized mandrel Schunk iTENDO on the trimming machine Activate the NTP service and the gateway Start the prediction model application on the gateway Start the data analysis platform and the operator HMI application Activate the data collection program of trimming machine process parameters Activate all the DMSs Start the work program for windows trimming on the trimming machine Access HMI and check the information displayed on the browser while the windows trimming occurs. When the trimming program is completed access the data analysis platform and check that data received and stored on it.
Expected output	The HMI shows the values of the trimming process variables measured by the DMS and the alerts raised by the prediction model (see 8.2.20). On the data analysis platform the parameters measured on the field during the trimming process are recorded and can be viewed (see 8.2.15).

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.2.22. Real-time execution of the prediction model

Test Name	Real-time execution of the prediction model
Test Type	System
Test purpose	Verify that the system is able to show the operator an alert in real time when there is a risk for the output quality due to a potentially dangerous situation of the measured parameters, so that he can decide to adjust the trimming machine settings through an override command.
Test input	Files with data streams produced by the various DMS nodes during the windows trimming on fuselage sections (see 8.2.14) and used for training and validate the prediction model (see 8.2.18)
Test description	Test prerequisites: • gateway connected to the network (cable), • validated prediction model deployed on the gateway (as containerized Azure application) • enterprise data analysis platform • HMI application • Web browser • stubs to feed the gateway with data streams on the input files Activate the gateway Start the prediction model application on the gateway Start the data analysis platform and the operator HMI application Launch the stubs to read the data streams from file and feed the gateway with it Access the operator HMI with a browser and check the information displayed on the browser.
Expected output	Verify that on average the HMI is able to show an alert within 2 seconds since a potentially dangerous situation of the process parameters occurs.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3. UC7 Test case details [LEONARDO]

8.3.1. Loading of aircraft failures data

Test Name	**Loading of aircraft failures data**
Test Type	Integration/Component
Test purpose	Verify that the Troubleshooting Component, through the Data Gathering is able to collect and store the aircraft faults data for the successive analysis/activities
Test input	Fault Data of a specific time period (last month of 2019)
Test description	-Open Data Upload portal -Upload ACAWS_FAILURE Table -Login as Maintenance Operator -Go to section with event list
Expected output	The page of the events list registered during flight show the faults of the last month of 2019 (see 8.3.7)

8.3.2. Loading of aircraft flight parameters

Test Name	**Loading of aircraft flight parameters**
Test Type	Integration/Component
Test purpose	Verify that the Troubleshooting Component, through the Data Gathering is able to collect and store the aircraft flight parameters for the successive analysis/activities
Test input	Aircraft flight parameters (Trend Data) of a specific time period (last month of 2019)
Test description	-Open Data Upload portal -Upload TREND_DATA Table -Login as Maintenance Operator -Go to the flight parameter vs time chart in event list section
Expected output	The page of trend parameters and event statistic charts shows the Flight parameter vs time chart also for the last month of 2019 (see8.3.8)

8.3.3. Loading of items removals

Test Name	**Loading of items removals**
Test Type	Integration/Component
Test purpose	Verify that the Troubleshooting Component, through the Data Gathering is able to collect and store the items removals for the successive analysis/activities

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Test input	List of items removed for maintenance (Removals) of a specific time period (last month of 2019)
Test description	-Open Data Upload portal -Upload REMOVALS Table -Login as Airframer Operator -Go to the page where the user can enter the investigation notes
Expected output	The page shows also the removals for the last month of 2019 (see 8.3.11)

8.3.4. Loading of troubleshooting manuals

Test Name	**Loading of troubleshooting manuals**
Test Type	Integration/Component
Test purpose	Verify that the Troubleshooting Component, through the Data Gathering is able to collect and store the troubleshooting manuals for the successive troubleshooting activities
Test input	Fault Isolation Manual (24FI)
Test description	-Open Data Upload portal -Upload Fault Isolation Manual -Login as Maintenance Operator -Go to possible solutions list for a failure relevant to aircraft system 24 -Open the manual
Expected output	Fault Isolation Manual for aircraft system 24 in .pdf available to follow the procedure: it is possible to open the manual from the section with possible solutions for a selected fault.

8.3.5. Troubleshooting component access by Maintenance Operator

Test Name	**Troubleshooting component access by Maintenance Operator**
Test Type	System/Component
Test purpose	Verify that a user with the Maintenance Operator role can gain access to the Troubleshooting features dedicated to that user profile.
Test input	
Test description	Login as a user with Maintenance Operator profile
Expected output	On login successful the user is enabled to: • Select an Aircraft and a Flight (progressive number) • Look at the list of the events of the chosen aircraft/flight • Look at trend parameters and event statistic charts • Analyse the faults and choose a solution from a list • Export data and reports

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3.6. Troubleshooting component access by Airframer Operator

Test Name	**Troubleshooting component access** by Airframer Operator
Test Type	System/Component
Test purpose	Verify that a user with the Airframer Operator role gains access to the Troubleshooting features dedicated to that role.
Test input	
Test description	Login as a user with Airframer Operator profile
Expected output	On successful login the user is enabled to: • Look at the list of removals and add investigation data • Look at analytics section • Export data and reports

8.3.7. List of fault events and event selection

Test Name	**List of fault events and event selection**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows the Maintenance Operator to look at a list of fault events for a chosen flight and to select the event to be investigated
Test input	-
Test description	- Login as a user with Maintenance Operator profile - Select an Aircraft and a Flight in the Maintenance Operator’s section - Go to page for the events list registered during flight
Expected output	The page shows a list of events and for each event: • Timestamp, combination of day and time at which the event was registered • Fault code • Event type (fault detected/recovered) • Flight phase, before flying, cruise, post flight • Average total maintenance time • Number of occurrences of this event in the last flight and in the last user-defined number of flights

8.3.8. Flight parameters chart

Test Name	**Flight parameters chart**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows the Maintenance Operator to look at flight parameters in a parameter vs time chart, to

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	change its scale and filter the parameters to be shown
Test input	-
Test description	- Login as a user with Maintenance Operator profile - Select an Aircraft and a Flight in the Maintenance Operator's section - Go to page with trend parameters and event statistic charts - Look at the flight parameters - Change the time scale - Select an additional flight parameter to be shown in the chart
Expected output	The page shows an Interactive flight parameters vs Time Chart for a selected flight

8.3.9. List of possible solutions for fault event

Test Name	**List of possible solutions for fault event**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows the Maintenance Operator, for each fault event, to open the Fault Isolation Manual and look at all the possible solutions to recover the fault, ordered by decreasing success rate, showing the most probable first
Test input	-
Test description	- Login as a user with Maintenance Operator profile - Select an Aircraft and a Flight in the Maintenance Operator's section - Select a Fault Event from section with the events list - Go to the section with possible solutions - Open Fault Isolation Manual - Look at possible solutions table
Expected output	The section shows the suggested solutions. For each solution the following information is displayed: • Part Number, an alphanumerical code that identify the item • Item description • Parts available at stock • Designed and actual Maintenance Elapsed Times and their deviation, calculated as the ratio between the designed and the actual times • Success rate, the percentage of times in which the reported solution allowed to recover the fault • Link to the Removal and Installation Procedures Manual • Failure causes statistics, which show the distribution of the defect root causes retrieved from the Airframer's investigation data

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3.10. Maintenance Operator Notes

Test Name	**Maintenance Operator Notes**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows the Maintenance Operator to enter notes concerning the maintenance performed on each fault event
Test input	
Test description	- Login as a user with Maintenance Operator profile - Select an Aircraft and a Flight in the Maintenance Operator's section - Select a Fault Event from the events list - Go to the section of possible solutions - Select the option to Add feedbacks - fill in the information related to: • Resources • Maintenance time • Maintenance operator skills • Other information concerning the performed maintenance operation.
Expected output	The maintenance feedback is shown in the troubleshooting optimization section for the Airframer Operator as described in 8.3.21

8.3.11. Airframer Notes

Test Name	**Airframer Notes**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows the Airframer Operator to insert notes relevant to the investigation performed on each removed item
Test input	
Test description	- Login as a user with Airframer profile - Open section for data investigation - Select a removal from the removals list - Select the option to enter notes relevant to the investigation performed - fill in the information related to: • Defect classification • Root cause • Root cause detail concerning the investigation on the of the removed item.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Expected output	The investigation Information is saved on the system and displayed for that removed item.

8.3.12. Association between failures and removals

Test Name	**Association between failures and removals**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component shows the association between fault events and removed items
Test input	-
Test description	- Login as a user with Airframer Operator profile - Go to Airframer Operator's section about analytics - Check the association list displayed between removals and fault events
Expected output	The list displayed shows associations of items removals and fault events, based on an algorithm that searches the faults occurred around the removal date of an item and then links a specific fault to the removal, by considering the specific aircraft model, the reference maintenance procedure in the manuals and the correlations detected by the analytics. The automatic associations can be validated in the Airframer Operator's section as described in 8.3.20

8.3.13. Correlation analysis between failures and flight parameters

Test Name	**Correlation analysis between failures and flight parameters**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows the Airframer Operator to look at possible correlations.
Test input	-
Test description	- Login as a user with Airframer Operator profile - Go to Airframer Operator's section about analytics
Expected output	In that section the system shows correlation matrixes between: • Flight parameter vs flight parameter • Fault vs fault • Fault vs flight parameter Each cell in the matrix shows the coefficient of correlation that expresses a measure of the strength of the relationship between the two variables. The automatic correlations can be validated in the Airframer Operator's section as described in 8.3.20.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3.14. Calculation of Investigation Statistics

Test Name	Calculation of Investigation Statistics
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component calculates Investigation Statistics based on the Airframer Operator notes
Test input	-
Test description	-Login as Airframer Operator -Go to Airframer Operator's section about analytics -Check the failure cause statistics calculated on investigation data
Expected output	For each removed item, the system displays failure causes statistics that show the distribution of the defect root causes entered by the Airframer operator (see Airframer operator notes in 8.3.11)

8.3.15. Maintenance Statistics visualization

Test Name	Maintenance Statistics visualization
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows the Airframer Operator to look at maintenance operations statistics.
Test input	-
Test description	-Login as a user with Airframer Operator profile -Go to Airframer Operator's section about troubleshooting optimization -Select a Fault Code -Look at the maintenance operations statistics concerning a fault code
Expected output	The system displays the list of the items removed for a fault along with: - success rate - actual Maintenance Elapsed Time and deviation with respect to the designed value

8.3.16. Export of maintenance activities report

Test Name	Export of maintenance activities report
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows to export a maintenance activity report for a selected flight or time period.
Test input	-

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Test description	- Login as a user with Maintenance Operator profile - Select an Aircraft and a Flight or period of time - Go to Maintenance Operator's section with events list - Export Maintenance Activity report
Expected output	The system produces a maintenance activity report on file, in .csv/.xlxs/.pdf, format that can be downloaded and saved and includes: - the fault events analysed - troubleshooting solution - actual maintenance time - maintenance Operator's feedbacks - removed items with details

8.3.17. Export of flight debrief report

Test Name	**Export of flight debrief report**
Test Type	System/Component
Test purpose	Verify that the Troubleshooting Component allows to export a flight debrief report for a selected flight.
Test input	-
Test description	-Login as a user with Maintenance Operator profile -Select an Aircraft and a Flight in the Maintenance Operator's section -Open section with events list -Export Flight Debrief report

Expected output	The system produces a Flight Debrief report on file in.csv/.xlxs/.pdf format, that can be downloaded and saved and includes:
	- Summary of the mission
	- Charts of the selected flight parameters vs time
	- Table with the list of fault events and relevant:
	    o Timestamp
	    o Fault code
	    o Event type (fault detected/recovered)
	    o Flight phase
	    o Average Total Maintenance time
	    o Number of occurrences of the event in the last flight and in the last user-defined number of flights

8.3.18. Troubleshooting

Test Name	**Troubleshooting**
Test Type	Acceptance
Test purpose	Verify that the Troubleshooting Component allows the Maintenance Operator to perform the troubleshooting activity by tracking the maintenance operations performed and recommending the most

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	successful possible solutions obtained from the elaboration of historical data and analytics correlations.
Test input	-
Test description	-Login as a user with Maintenance Operator profile -Select an Aircraft and a Flight **1-Fault Debriefing** -Open Maintenance Operator's section with events list -Look at the events list and choose a Fault Code **2-Fault Isolation and solution identification** -Open section with possible solution -Open Fault Isolation Manual -Select a proposed solution - After the maintenance intervention, add feedback (Maintenance operator's notes) **3-Flight and Maintenance reports exporting** -Export Flight Debrief report -Export Maintenance Activity report
Expected output	1. Events list with relevant timestamp, fault code, event type (fault detected/recovered), flight phase, Average Total Maintenance time, number of occurrences of this event in the last flight and in the last user-defined number of flights 2. Possible solutions list with relevant part number, description, parts available at stock, average/design maintenance time maintenance time deviation and success rate. 3a. Flight Debrief report on file (see 8.3.16) 3b. Maintenance activity report on file (see 8.3.17),

8.3.19. Investigation Data

Test Name	**Investigation Data**
Test Type	Acceptance
Test purpose	Verify that the Troubleshooting Component allows the Airframer Operator to add the results of dedicated post-removal failure investigations to provide a feedback between the fault and the root cause of the defect, in order to improve the troubleshooting and avoid no fault found events.
Test input	-
Test description	- Login as a user with Airframer Operator profile - Go to Airframer Operator's section about the investigation data - Filter the data by description and/or aircraft and select a removal - Insert notes relevant to the investigation performed on the removed

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	item
Expected output	Investigation performed fields (Defect classification, Root cause, Root cause detail) filled in the relevant removal of the removals list The investigation Information entered (Defect classification, Root cause, Root cause detail see 8.3.11) is saved on the system and used to calculate the investigation cause statistics (see 8.3.14).

8.3.20. Analytics

Test Name	**Analytics**
Test Type	Acceptance
Test purpose	Verify that the Troubleshooting Component allows the Airframer Operator to use analytics models to identify correlations, patterns and statistics from the aircraft and maintenance generated data. The results of these models, once validated, can be exported to support the Maintenance Operator in his troubleshooting activities.
Test input	-
Test description	- Login as a user with Airframer Operator profile - Go to Airframer Operator's analytics section **1-Data analysis** - Select a Fault Code - Look at the results in the correlation matrix and choose a correlation to investigate -Look at the associations between fault event and removals proposed by the system -Look at the failure cause statistics calculated based on investigation data **2-Test and validation** -Test the effects of the selected correlation -Validate the correlation **3-Export the model** -Export the updated correlation table
Expected output	1a. Correlation Matrix between: o Flight parameter vs flight parameter o Fault vs fault o Fault vs flight parameter 1b. Association list between items removals and fault events 1c. For each item, the failure causes statistics that show the distribution of the defect root causes based on Airframer investigation data 2. Updated success rates, possible solutions and investigations statistics and comparison with the old ones 3. Validated values are used in the Troubleshooting Component in

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	order to calculate and show the success rates, possible solutions and the investigations statistics in the Maintenance Operator's section

8.3.21. Troubleshooting Optimization support

Test Name	**Troubleshooting Optimization support**
Test Type	Acceptance
Test purpose	Verify that the Troubleshooting Component allows the Airframer Operator to export Troubleshooting and Analytics results to support Troubleshooting optimizations activities that are performed outside the system
Test input	-
Test description	- Login as Airframer Operator - Go to Airframer Operator's troubleshooting optimization section - Select a Fault Code - Export data
Expected output	For the selected Fault Code the system produces and allows to download a report on .csv/.pdf file that contains: • List of the item removed with: part number, success rates and Maintenance Time Deviation • Validated correlations list between failures and flight parameters, failure cause statistics based on investigation data • maintenance notes

8.3.22. Identification of Valid Correlations

Test Name	**Identification of Valid Correlations**
Test Type	Acceptance
Test purpose	Verify that the Troubleshooting Component allows the Airframer Operator to perform data analysis on failure and flight parameters in order to identify known or new valid correlations through the analytics models
Test input	A set of Fault codes and items removals to be investigated
Test description	- Login as a user with Airframer Operator user profile - Go to Airframer Operator's analytics section - Select a Fault Code - Look at the results in the correlation matrix - Look at the automatic association between fault events and removals - Repeat the process for the fault codes and items removals to be

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	investigated.
Expected output	The system calculates: • correlations coefficients between failures and failures vs flight parameters • association between fault events and removals that match the expected results deriving from reference engineering data, for known correlations, or that are confirmed with the support of engineering specialists and empirical data, in case of new correlations

8.3.23. Collection of warehouse data

Test Name	**Collection of warehouse data**
Test Type	Integration/Component
Test purpose	Verify that the Spare Management Component, through the Data Gathering is able to collect and store the warehouse data for the successive analysis/activities
Test input	Warehouse data in a specific time period (last month of 2019)
Test description	- Login as Administrator Operator - Open Data Upload portal -Upload STOCK_QTY, WAREHOUSE_ENV, WAREHOUSE_ENV_THR Tables -Login as a user with Logistic Operator profile -Go to warehouse data section - export parts availability report (see 8.3.37)
Expected output	The parts availability report shows stocks availability taking into account the data of the last month of 2019; The Stock optimization page (see 8.3.44) shows the available stocks taking into account the data of the last month of 2019

8.3.24. Collection of warehouse in/out tracking data

Test Name	**Collection of warehouse in/out tracking data**
Test Type	Integration/Component
Test purpose	Verify that the Spare Management Component, through the Data Gathering is able to collect and store the warehouse in/out tracking data to support the spare management analysis
Test input	Warehouse in/out tracking data in a specific time period (last month of 2019)
Test description	- Login as Administrator Operator - Open Data Upload portal - Upload SAP_TT, SAP_PEGGING Tables

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	- Login as a user with Logistic Operator user profile - Go to item tracking data section in the Stock optimization page
Expected output	The Stock optimization page (see 8.3.44) shows the in/out stock movements occurred in the last month of 2019

8.3.25. Collection of flight activity data

Test Name	**Collection of flight activity data**
Test Type	Integration/Component
Test purpose	Verify that the Spare Management Component through the Data Gathering is able to collect and store the flight activity data (planned flight hours per aircraft per day) to support the spare management analysis
Test input	List of planned flight activity data (last month of 2019)
Test description	- Login as Administrator Operator -Open Data Upload portal -Upload AIRCRAFT_FH_FORESEEN Tables -Login as a user with Airframer Operator profile -Go to the stock optimization section
Expected output	The planned Flight hours displayed in the stock optimization section (see 8.3.44) takes into account the planned flight hours of the last month of 2019.

8.3.26. Spare Management component access by Logistic Operator

Test Name	**Spare Management component access by Logistic Operator**
Test Type	System/Component
Test purpose	Verify that a user with the Logistic Operator role gains access to the Spare management features dedicated to that role.
Test input	
Test description	Login as a user with Logistic Operator profile
Expected output	On Login successful the user is enabled to: • Look at the actual stock status in terms of quantity and relevant indicators • Look at the most unreliable items and their reliability indicators • Look at items subjected to scheduled maintenance with relevant estimated expiration date and scheduled maintenance peculiarities • Look at tracking status of removed parts sent for repairs or new parts ordered • Register Aircraft On Ground (AOG) events due to missing parts.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3.27. Spare Management component access by Airframer Operator

Test Name	**Spare Management component access by Airframer Operator**
Test Type	System/Component
Test purpose	Verify that a user with the Airframer Operator role gains access to the Spare management features dedicated to that role.
Test input	
Test description	-Login as a user with Airframer Operator profile
Expected output	On successful login, the user is enabled to: • Look at items stock status with relevant indicators • Look at tracking status of removed parts sent for repairs • Look at flight hours status

8.3.28. Scheduled maintenance activities expiration date

Test Name	**Scheduled maintenance activities expiration date**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component allows the Logistic Operator to look at parts subject to scheduled and calculates the Estimated Expiration Date when the scheduled maintenance should take place
Test input	-
Test description	- Login as a user with the Logistic Operator profile - Go to section about scheduled maintenance activities - Check the listed items in the scheduled maintenance list and the Estimated Expiration Date that the system calculated.
Expected output	The system shows a list of parts that are scheduled for maintenance and shows the Estimated Expiration Date when the item must be removed for maintenance that is calculated considering: • the remaining flight hours before the next scheduled maintenance based on the actual flight hours to the date • the planned daily flight hours Parts that have closer Estimated expiration Date are displayed first in the list.

8.3.29. Reliability indicators

Test Name	**List of top unreliable items**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component allows the Logistic

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	Operator to analyse what are the most unreliable parts based on the Reliability KPIs of those parts
Test input	-
Test description	- Login as a user with the Logistic Operator profile - Go to section about Top Unreliable Items - set an observation interval and selection criteria for the items to be displayed - Check the items displayed and the Reliability KPIs that the system shows for them.
Expected output	The system shows a list of parts grouped by item Reference ID, and for each of them the following indicators are displayed: • MTBUR (Mean Time Between Unscheduled Removals), the average time between two unscheduled removals. Calculated considering the number of flight hours achieved, the item quantity installed on aircraft and the unscheduled removals, which are the removals due to corrective maintenance • URR (Unscheduled Removal Rate), reciprocal of the MTBUR expressed in 1000 flight hours • GRADIENT, angular coefficient of the linear regression equation of the URR evolution over an observation interval • STANDARD DEVIATION, standard deviation of the URR values with respect to the average URR over the observation interval.

8.3.30. Updating of Reliability KPI

Test Name	**Updating of Reliability KPI**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component automatically updates the Reliability KPI depending on the amount of aircraft flight hours loaded in the system
Test input	
Test description	- Check preliminarily the Reliability KPIs of a selection of items (see 8.3.29) - Load new flight parameters data (See test case 8.3.25) - Login as a user with Logistic Operator profile - Open Logistic Operator's section with top unreliable items list - Check the items displayed and the Reliability KPIs that the system displays for that selection of items
Expected output	The Reliability KPIs of the selected items are updated according to the new flight hours loaded in the system.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3.31. Availability Warning and relevant performance indicators

Test Name	**Availability Warning and relevant performance indicators**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component displays the Availability Warning for the unreliable item and the Performance indicators it is based on.
Test input	-
Test description	- Login as a user with Logistic Operator profile - Open Logistic Operator’s section of stock status - set selection criteria for the items to be displayed - check the availability warning displayed for the selected items - Choose one of the items displayed and check the performance indicators for that item
Expected output	The system shows a list of parts grouped by item Reference ID, and for each item in the list it displays an Availability Warning (green, yellow or red) on possible spare parts issues, calculated as a weighted average of three performance indicators (Failure Pattern Detector, Removal Rate Alert, Risk of Shortage). For the selected item in the list the system shows these indicators: - Failure Pattern Detector, indicates a criticality due to recurring faults which may lead to an item removal, as derived from the Troubleshooting Components analytics and from reference manuals - Removal Rate Alert, indicates risky deviations of the URR from the its average values, which represent a potential critical increment of the removal rate of the item - Risk of Shortage, indicates the probability of not having parts at stock when requested. Each indicator is calculated as a numerical value between 0 and 100, and has a relevant weight to be used in the weighted average method that allows to obtain the Availability Warning

8.3.32. Modification of weights and thresholds

Test Name	**Modification of weights and thresholds**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component allows the Logistic Operator to change the weights for the Availability Warning calculation and the thresholds concerning the performance indicators
Test input	
Test description	- Login as a user with Logistic Operator profile - Check preliminarily the availability warning and Performance Indicators for the items of a selected part nr. (see 8.3.31)

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	-Open the section to set weights and thresholds - select that part nr. - Change performance indicators weights for the availability warning on that part nr. - Open the section to set warnings thresholds - Change the thresholds of performance indicators for that part nr - Check again the availability warning and Performance Indicators for the items of a selected part nr.
Expected output	The availability warning and Performance Indicators (Failure Pattern Detector, Removal Rate Alert, Risk Of Shortage) for the items of the part nr. are now calculated according to the new weights and thresholds entered.

8.3.33. Insert of AOG event

Test Name	**Insert of AOG event**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component allows the Logistic Operator to document if an item removal has caused an Aircraft On Ground (AOG) event due to missing parts
Test input	
Test description	- Login as a user with Logistic Operator profile - Go to Logistic Operator’s administrative section on item removals - filter removals according to specific selection criteria - Check/Uncheck the column to say if a removal in the list caused an AOG event
Expected output	Open the section on top unreliable items and verify that for the items displayed the AOG column counts also the new AOG events.

8.3.34. Recommendation of weights and thresholds

Test Name	**Recommendation of weights and thresholds**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component suggests to the Logistic Operator recommended weights for the Availability Warning and thresholds for the performance indicators
Test input	-
Test description	- Login as a user with Logistic Operator profile - Go to Logistic Operator’s section with stock status - Select an item where AOG events occurred and have been registered

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	(see 8.3.33) - Open the section to set warnings weights - Check the recommended values displayed beside each weight - Open the section to set the thresholds of performance indicators - Check the recommended values displayed beside each threshold - apply the recommended values of weights and thresholds for that item (see 8.3.32) - check the availability warnings for that item
Expected output	The system shows recommended values beside each weight and thresholds. Recommended values are suggested from the analysis of historical data on: • Removals • Fault associated to removals • AOG events • Reliability KPI so to have warnings to reduce the risk of AOG events. The availability warnings for the selected item reflect the AOG events occurred that have been registered for that item.

8.3.35. Visualization of performance indicators

Test Name	**Visualization of performance indicators**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component displays in a graphical form to the Logistic Operator the calculated performance indicators
Test input	-
Test description	- Login as a user with Logistic Operator profile - Go to the section about Top Unreliable Items - chose one of the items displayed and open the section showing the performance indicators for that item - check how the performance indicators are displayed
Expected output	• The Failure Pattern Detector is displayed with a gauge-like representation, using different colours (green, yellow, red) to discriminate between low, medium and high criticality; • Removal Rate Alert is displayed as a line chart that shows the evolution of the URR over an observation period, compared with two lines predefined alerts levels • Risk Of Shortage is displayed as pie, where the red slice indicates the risk of shortage (ROS) and the green slice the probability to have available parts when requested

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3.36. Recommended stock size

Test Name	**Recommended stock size**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component calculates and shows to the Logistic Operator the recommended stock size
Test input	-
Test description	- Login as a user with Logistic Operator profile - Open the section about stock status - Choose a part number in the list - Check the recommended stock size for that item
Expected output	For the selected part the system shows the recommended quantity in stock that is obtained with an iterative optimization method to meet the target ROS over a predefined period of time. The optimization considers: • the target ROS to be satisfied (e.g. ≤ 5%) • the planned flight hours • the MTBUR • the quantity of parts installed on aircraft • the target period of time where the target ROS must be achieved

8.3.37. Export of parts availability report

Test Name	**Export of parts availability report**
Test Type	System/Component
Test purpose	Verify that the Spare Management Component allows to export a parts availability report
Test input	-
Test description	-Login as a user with Logistic Operator profile -Open Logistic Operator’s section about stock status -Export the Parts availability report
Expected output	The system produces a Parts Availability report on file in.csv/.xlsx/.pdf format, that can be downloaded and saved and for each item indicates: • Part description • Part Number • Quantity available at stock • Position of item spare parts in the warehouse • Recommended stock size • Number of parts under repair • Number of new parts ordered • Availability warning

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.3.38. Export of scheduled activities report

Test Name	Export of scheduled activities report
Test Type	System/Component
Test purpose	Verify that the Spare Management Component allows exporting a scheduled activities report containing all the items with scheduled maintenance activities.
Test input	-
Test description	- Login as a user with Logistic Operator profile - Open the section about scheduled maintenance activities - Export the scheduled activities report
Expected output	The system produces a Scheduled Maintenance Report on file in .csv/.xlsx/.pdf format, that can be downloaded and saved and for each item indicates: • Part Number • Serial Number • Part Description • Aircraft • Tasks to be performed • Remaining flight hours to scheduled maintenance • Estimated Expiration Date • Availability warning

8.3.39. Export top unreliable items report

Test Name	Exporting top unreliable items report
Test Type	System/Component
Test purpose	Verify that the Spare Management Component allows to export a report for a selected number of top unreliable items.
Test input	-
Test description	- Login as a user with Logistic Operator profile - Go to section about Top Unreliable Items - set selection criteria for the items to be displayed and the number of items to be shown - Export top unreliable items report
Expected output	The system produces a report for Top Unreliable Items report on file in.csv/.xls/.pdf format that can be downloaded and saved, where for each item the following information is indicated: • Part description • Reference ID • MTBUR • URR

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	• GRADIENT • STANDARD DEVIATION • Number of AOCP events • Availability Warning

8.3.40. Stock demand

Test Name	**Stock demand**
Test Type	Acceptance
Test purpose	Verify that the Spare Management Component allows the Logistic Operator to monitor the items at stock and see if there is a demand to increase those items (now orders)
Test input	-
Test description	- Login as user with Logistic Operator profile - Go to Logistic Operator's section about new orders - Check the list of items at stock and the information displayed - Export the Parts availability report
Expected output	The system shows the list of items at stock and for each item the following information is displayed: • Part number • Relevant aircraft system • Quantity at stock • Recommended stock size • Demand (as the difference between recommended stock size and the quantity at stock) • Parts at repair or ordered • Availability Warning The system produces a Parts availability report on file in .xlsx / .csv / .pdf format that can be downloaded and saved (see 8.3.37).

8.3.41. Top Unreliable Items monitoring

Test Name	**Top Unreliable Items monitoring**
Test Type	Acceptance
Test purpose	Verify that the Spare Management Component allows the Logistic Operator to monitor the Top Unreliable Items and supports him showing reliability indicators and Availability Warnings
Test input	-
Test description	- Login as user with Logistic Operator profile - Go to the section about Top Unreliable Items - Select an observation interval and the number of items to be displayed - Check the items displayed and the Reliability KPIs that the system

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	displays - order them by increasing URR - Go the section about stock status - Check the recommended stock size for those items - Export the Top Unreliable Items report
Expected output	The system shows a list of parts grouped by item Reference ID, and for each of them the following information is displayed: • Part description • Reference ID • MTBUR (see 8.3.29) • URR (see 8.3.29) • GRADIENT (see 8.3.29) • STANDARD DEVIATION (see 8.3.29) • Number of AOCP events • Availability Warning (see 8.3.31) • Parts at repair or new orders The system shows the recommended quantity in stock for the selected items (see 8.3.36) The system produces a report for Top Unreliable Items report on file in.csv/.xlsx/.pdf format that can be downloaded and saved (see 8.3.39).

8.3.42. Scheduled Maintenance monitoring

Test Name	**Scheduled Maintenance monitoring**
Test Type	Acceptance
Test purpose	Verify that the Spare Management Component allows the Logistic Operator to monitor the Scheduled Maintenances performances and supports him in the decision making process to guarantee the necessary logistic support for the scheduled maintenance activities
Test input	-
Test description	- Login as a user with the Logistic Operator profile - Go to section about scheduled maintenance activities - Check the listed items in the scheduled maintenance list. - Export Scheduled Maintenance report
Expected output	The system shows a list Items subject to scheduled maintenance ordered by increasing Estimated Expiration Date, and for each of them the following information is displayed: • Part number • Serial number • Description • Aircraft • Aircraft system • Task to be achieved • Remaining flight hours to scheduled maintenance • Estimated Expiration Date (see 8.3.28)

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	• Availability warning (see 8.3.31) The system produces a Scheduled Maintenance Report on file in.csv/.xlsx/.pdf format, that can be downloaded and saved (see 8.3.38).

8.3.43. Tuning of warnings

Test Name	**Tuning of warnings**
Test Type	Acceptance
Test purpose	Verify that the Spare Management Component allows the Logistic Operator to register new AOG events for missing parts (AOCP) and change the weights and threshold of the Availability Warning to better identify the items on which corrective and preventive logistic support actions are necessary.
Test input	-
Test description	- Login as a user with Logistic Operator profile - Go to Logistic Operator's administration section - Go to Logistic Operator's administrative section on item removals - filter removals according to specific selection criteria a) Check the AOCP column of a removal that caused an AOG event - Open the section to set weights and thresholds - select that part nr. that caused the AOG event - set the recommended values (see 8.3.34) for weights of performance indicators in the availability warning on that part nr. - Open the section to set warnings thresholds - Set the recommended values (see 8.3.34) for the thresholds of the performance indicators for that part nr. b) Check the availability warning and Performance Indicators for that part.
Expected output	a) the AOG column for the selected item displayed in section on top unreliable items counts also the new AOG event entered b) The availability warnings displayed in the stock status are updated according to the new values set (see 8.3.32) and reflect the AOG event occurred registered for that item.

8.3.44. Stock status monitoring for optimization

Test Name	**Stock status monitoring for optimization**
Test Type	Acceptance
Test purpose	Verify that the Spare Management Component allows the Airframer Operator to obtain information and indicators derived from field activities to support the optimization of the logistic support services offered to the end user.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Test input	-
Test description	- Login as a user with Airframer Operator profile - Go to the section about stock optimization - Check the items displayed and choose an item - Check the performance indicators for the selected item - Check the items at repair or new orders - check aircraft flight activity information
Expected output	The system shows a list of items and for each of them the following information is displayed: • part number, • description, • quantity at stock, • recommended stock size • AOG events. For the selected item the system displays both in graphical and in numerical forms these performance KPIs: • Failure Pattern Detector (see 8.3.31, 8.3.35) • Removal Rate Alert (see 8.3.31, 8.3.35) • Risk Of Shortage (see 8.3.31, 8.3.35) The system shows a the list of items at repair that can be fitered by part nr, serial nr, status of repair and shows tracking information on each item at repair.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.4. UC8 Test case details [TRUMPF]

8.4.1. Semantic Enrichment Module Test

Test Name	Semantic Enrichment Module Test
Test Type	component
Test purpose	Testing whether the detection and recognition rate of TRUMPF and third-party machines from 2D images and a 3D shopfloor scan is sufficiently high. The test success metrics are defined in D3.3.
Test input	2D images and 3D shopfloor scan from TRUMPF customer center
Test description	The semantic enrichment module is tested both on 2D and 3D data. The 2D image detection and recognition is tested on a labelled dataset, which was collected by acs plus. The performance is measured by accuracy. The 3D technology of the semantic enrichment module receives 2D images and 3D hall scans. The results are compared with the actual shopfloor plan of the TRUMPF customer center. The following metrics are evaluated: • Accuracy: Which percentage of machines is identified correctly • Intersection over Union (IoU): How large is the overlap between the Bounding Boxes predicted by the models and the real Bounding Boxes given by the annotated shop floor? Currently there is no customer shopfloor scan available that can be used for the validation of the recognition rate of third-party machines. Using the results of the first shopfloor scan provider was technically not feasible because the 2D-images could not be extracted and the quality of the pointcloud was not satisfactory as well.
Expected output	The performance of the 2D technologies on the 2D images should achieve 80% accuracy. It can be demonstrated that the 3D technologies can achieve 80% IoU.

8.4.2. UWB Infrastructure Test

Test Name	UWB Infrastructure Test
Test Type	component
Test purpose	Test if the UWB Infrastructure works as expected and in the defined operation limits
Test input	Installed Track and Trace system in a shop floor. Instrumentation of the system following our test plan which includes: simultaneous movement of a larger number of tags, benchmarking of the system (comparing real and estimated positions), associating product/order information to tags. Uploading position information to cloud data center.
Test description	UWB infrastructure is deployed and evaluated based on the Track and Track release qualification tests. This includes a localization quality assessment and cloud connectivity testing, among others.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

Expected output	Positioning performance is within specified boundaries. Position and order information successfully communicated to cloud data center for further processing. UI and Hardware E2E all work as expected and defined.

8.4.3. Interface Test

Test Name	Interface Test
Test Type	integration
Test purpose	Validation whether the results from the semantic enrichment module are exported correctly in the defined data exchange format. The results should then be importable to the simulation model framework.
Test input	machine types and respective positions from 8.4.1
Test description	The results from 8.4.1 are exported in the defined exchange format. The exchange file is checked for compliance with the .xml standard and the defined structure. It this then checked, if the results can be imported into the simulation model.
Expected output	The export from the semantic enrichment module results into the defined .xml exchange format works properly and the results can be imported into the simulation model framework.

8.4.4. Simulation Model Unit Tests

Test Name	Simulation Model Unit Test
Test Type	component
Test purpose	Verification whether the simulation model units interact correctly with each other.
Test input	Updated simulation model library Test production orders
Test description	For each release of the simulation model library automatic test cases are performed as depicted in Figure 7. Their results are exported as an Excel file shown in Figure 8. In each test cases different combinations of machines, automation units and intralogistics agents like AGVs or workers are performed. It is tested whether products reach their expected destinations.
Expected output	Findings on errors that occurred due to an update of the simulation model logic.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

```
@ECHO OFF
rem
rem Copy Excel File Template
copy "%cd%\ModelTestLibrary\*.xlsx" "%cd%\ExportedTests"
rem
rem
rem Copy xml files needed for OrderSchedulerTests
rem
copy "%cd%\OrderSchedulerTests\*.xml" "%cd%\ExportedTests\OrderSchedulerTests_RunOrderSchedulerTest"
copy "%cd%\OrderSchedulerTests\*.xml" "%cd%\ExportedTests\OrderSchedulerTests_RunWorkOrderSourceTest"
cls
rem
rem  RunAnylogicTest.bat        TestFolderName      TestCaseName          PackageName          TestModelleDir.
rem  -----------------------    -----------         --------------------- -------------------  --------------
call RunAnylogicTest.bat        EdgeBreakerTest     RunEdgeBreakerTest    edgebreakertest      %cd%
call RunAnylogicTest.bat        FlatMasterTest      RunFlatMasterTest     flatmastertest       %cd%
call RunAnylogicTest.bat        LiftMasterTest      RunLiftMasterTest     liftmastertest       %cd%
call RunAnylogicTest.bat        LiftMasterTest      RunLiftMasterStoreTest liftmastertest      %cd%
call RunAnylogicTest.bat        LoadMasterAK10Test  RunLoadMasterAK10Test loadmasterAK10test   %cd%
```

Figure 7 Batch file that executes the automatic unit tests

Creation Date: 03.11.2020 02:51:31

Test Model	Experiment	TestCase	TestCase Description	Check Description	CheckResult
EdgeBreakerTest	RunEdgeBreakerTest				
		Test1_testCase	Testet die korrekte Verarbeitungvon Jobs und Parts.Prüfung der Visualisierung in geänderter Orientierung.		
			Test_automaticCheck1	Nach 18 Minuten muss der erste Job und 10 Parts verarbeitet sein.	pass
			Test_automaticCheck2	Wenn der Job beendet ist müssen 10 Parts verarbeitet sein.	pass
			Test_automaticCheck3	Check if height, width, length and weight are set correctly	pass
			Test_automaticCheck4	Check if operationID is set correctly	pass
		Test2_testCase	Testet die korrekte Verarbeitungvon Jobs und Partsbei automatisiertem Beladen und geänderter Orientierung.		
			Test2_automaticCheck1	Nach 18 Minuten muss der erste Job und 10 Parts verarbeitet sein.	pass
			Test2_automaticCheck2	Wenn der Job beendet ist müssen 10 Parts verarbeitet sein.	pass
		Test3_testCase	Testet das korrekte Zusammenspielmit einem Lagerbaustein bezüglich Materialanforderung. Unter Anderemob Folgepaletten für einen Auftragangefordert werden.		
			Test3_automaticCheck1	Nach 40 Minuten muss der erste Job und 18 Parts verarbeitet sein.	pass
			Test3_automaticCheck2	Wenn der Job beendet ist muss die zweite Palette weg sein.	pass
		Test4_testCase	Testet den Bürstenwechselmechanismusmit unterschiedlichen Materialien und nach Zeit.		
			Test4_automaticCheck1	Zwischen Job 1 und Job 2 findet ein Bürstenwechsel wegen Material- wechsel statt.	pass
			Test4_automaticCheck2	Zwischen Job 2 und 3 findet ein zeitbedingter Bürstenwechsel statt.	pass
FlatMasterTest	RunFlatMasterTest				
		Test1_testCase	Testet die korrekte Verarbeitungvon Jobs und Parts.Prüfung der Visualisierung in geänderter Orientierung.		
			Test1_automaticCheck1	Nach 18 Minuten muss der erste Job und 10 Parts verarbeitet sein.	pass
			Test1_automaticCheck2	Wenn der Job beendet ist müssen 10 Parts verarbeitet sein.	pass
			Test1_automaticCheck3	Check if height is set correctly	pass
			Test1_automaticCheck4	Check if operationID is set correctly	pass

Figure 8 Export of the unit test results as Excel file

8.4.4.1. Simulation Model Performance Test

Test Name	Simulation Model Unit Test
Test Type	component
Test purpose	Verification whether the simulation model units interact correctly with each other.
Test input	Updated simulation model library Test production orders
Test description	For each release of the simulation model library automatic test cases are performed. In each test cases different combinations of machines, automation units and intralogistics agents like AGVs or workers are performed. It is tested whether products reach their expected destinations.
Expected output	Findings on errors that occurred due to an update of the simulation model logic.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.4.5. Simulation Model Generation Test

Test Name	Simulation Model Generation Test
Test Type	system
Test purpose	This test shows if a) data from UWB infrastructure and Semantic Enrichment Module can be received and interpreted by the Simulation Model Framework and b) if this data can be used to produce an initial Simulation Model
Test input	- Machine positions from Semantic Enrichment (either using cloud, offline or direct upload) - UWB localization data (either using cloud, offline or direct upload)
Test description	- A shop floor scan is performed and processed by the Semantic Enrichment Module. The extracted data is provided to the Simulation Model Framework - Locations and location-bound order information is provided to the Simulation Model Framework - Simulation Model Framework consumes this data - Simulation Model Framework creates an initial model based on this data - The created model is evaluated by an expert
Expected output	The created model meets the requirements for an initial Simulation Model.

8.4.6. Overall Use Case Test

Test Name	Overall Use Case Test
Test Type	acceptance
Test purpose	System validation
Test input	3D shopfloor scan and 2D images from TRUMPF customer centre
Test description	The semantic enrichment module detects and recognizes the machines and their respective positions from the 3D shopfloor scan and 2D images. This information is exported in the defined exchange format which is fed into the simulation model generator. The resulting simulation model is compared to a reference model that has been created manually according to the existing floor plan.
Expected output	executable material flow simulation model of the TRUMPF customer centre

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

8.5. UC9 Test case details [WIKA]

8.5.1. Simple Lift

Test Name	Simple lift
Test Type	System
Test purpose	The show what kind of information shall be provided to the crane operator.
Test input	Lift the load straight up by 5m.
Test description	The MATLAB-simulation will lift the object by 5m and provide a video stream of that to the drone simulation. The drone will measure the position of the load. The difference can be used for process calibration.
Expected output	The deviation between the two simulations should be less than 10%.

8.5.2. 2 axis lift

Test Name	2 axis lift
Test Type	System
Test purpose	The show what kind of information shall be provided to the crane operator.
Test input	Lift the load straight up by 5m. Wait 5s. Then move the load 3m to the left.
Test description	The MATLAB-simulation will lift the object as described in "Test input" and provide a video stream of that to the drone simulation. The drone will measure the position of the load. The difference can be used for process calibration.
Expected output	The deviation between the two simulations should be less than 10%.

8.5.3. 3 axis lift

Test Name	3 axis lift
Test Type	System
Test purpose	The show what kind of information shall be provided to the crane operator.
Test input	Lift the load straight up by 5m. Wait 5s. Then move the load 3m to the left.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276

	Wait 5s. Then move the load 2m forward.
Test description	The MATLAB-simulation will lift the object as described in “Test input” and provide a video stream of that to the drone simulation. The drone will measure the position of the load. The difference can be used for process calibration.
Expected output	The deviation between the two simulations should be less than 10%.

8.5.4. 3 axis and rotation lift

Test Name	3 axis and rotation lift
Test Type	System
Test purpose	The show what kind of information shall be provided to the crane operator.
Test input	Lift the load straight up by 5m. Wait 5s. Then move the load 3m to the left. Wait 5s. Then move the load 2m forward. Wait 5s. Then rotate the load by 30° clock wise.
Test description	The MATLAB-simulation will lift the object as described in “Test input” and provide a video stream of that to the drone simulation. The drone will measure the position of the load. The difference can be used for process calibration.
Expected output	The deviation between the two simulations should be less than 10%.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276