D8.8 – FINAL VALIDATION RESULTS

Deliverable ID	**D8.8**
Deliverable Title	**Final Validation Results**
Work Package	**WP8**
Dissemination Level	**PUBLIC**
Version	**1.1**
Date	**04-01-2020**
Status	**Final**
Lead Editor	**SLAB**
Main Contributors	**Ákos Milánkovich (SLAB), Judit Torma (SLAB)**

Published by the CPSwarm Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 731946.

Document History

Version	Date	Author(s)	Description
0.1	2019-09-27	Ákos Milánkovich (SLAB)	First Draft with TOC and imported test cases from D2.8 and D8.7, added new tests for new requirements.
0.2	2019-12-05	Ákos Milánkovich (SLAB)	Added use-case relevant content, added FIT contributions.
0.3	2019-12-18	Ákos Milánkovich (SLAB)	Added comments for fields to be filled, integrated feedback
0.4	2019-12-20	Ákos Milánkovich (SLAB)	Integrated partner contributions, conclusions
0.5	2019-12-23	Judit Torma (SLAB)	Internal review and modifications
1.1	2020-01-04	Judit Torma (SLAB)	Integrated result of partner reviews, preparing final version to be submitted

Internal Review History

Review Date	Reviewer	Summary of Comments
2019-12-30	Arthur Pitman (UNI-KLU)	Corrected typos and updated test cases
2020-01-02	Etienne Brosse (SOFTEAM)	Chapter structure and summary table

Executive Summary

This deliverable reports the results of the final set of evaluations performed on the CPSwarm components that was obtained by applying the methodology and tests defined in Deliverable D2.8 – Validation Framework Specification to the state of the CPSwarm components as of the end of Phase 3. CPSwarm components were evaluated with a combination of test cases and Key Performance Indicator (KPI) thresholds.

The summary of the evaluation results is as follows:

- Modelling Tool: TRL 5
- Modelling Library: TLR 5
- Optimization Tool: TRL 4
- Simulation Tool: TRL 5
- Code Generation Tool: TRL 5
- Deployment Tool: TRL 5
- Hardware Abstraction Layer: TRL 5
- Monitoring Tool: TRL 5

Following the validation framework specification defined in Task 2.4, in this task the relevance of requirements to specific use cases is also presented.

Table of Contents

1 Introduction

1.1 Validation activities

Since its inception, the CPSwarm project has been striving to create a workbench that is relevant to the current state-of-the-art of the industry and that can offer a solution that is more effective and integrated than available alternatives. To achieve this, an iterative methodology was developed to advance the toolset, as depicted in Figure 1.

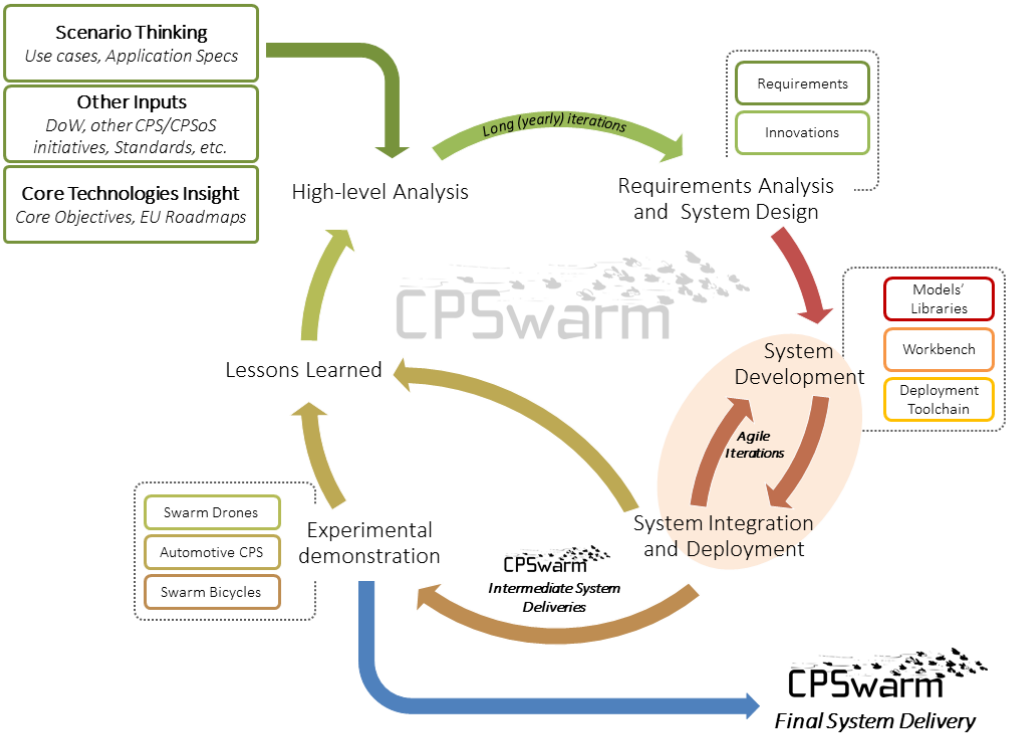

Figure 1 – CPSwarm project lifecycle

The development process integrates validation and verification into the project lifecycle in a similarly iterative manner (see Figure 2)

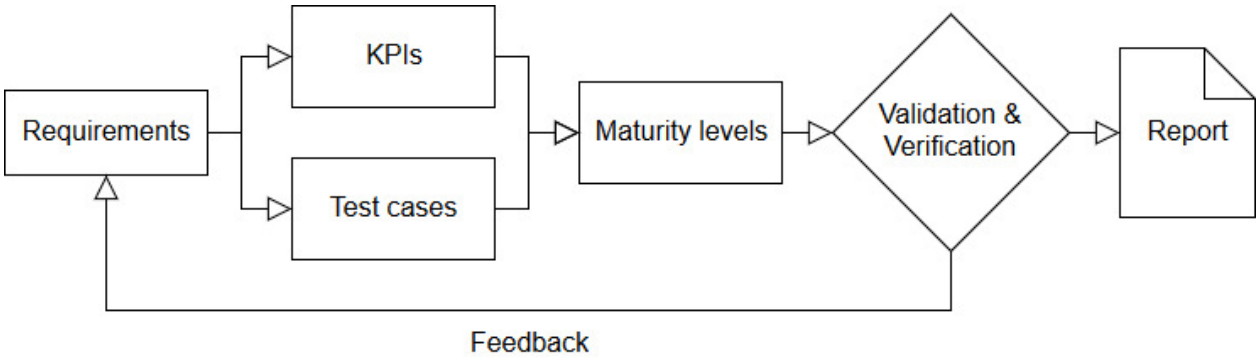

Figure 2 – Iterative feedback cycle in CPSwarm

After the first stages of requirements engineering were completed [1] [2], a rich set of test cases and key performance indicators were defined to help measure the progress of the project towards meeting the

requirements that had been set. These test cases and key performance indicators were documented [3] alongside a methodology determining the overall project maturity as part of the Validation Framework.

1.2 Measurement and reflection

The iterative nature of the project methodology made it necessary to periodically evaluate, revise and reengineer the requirements and relevant test cases and key performance indicators as the project was progressing. This report aims to give a realistic view on the final state of the project, for internal and external stakeholders alike. While test cases and KPIs for existing requirements are expected to be static, new requirements may have been created as more advanced functionality was developed in the prototype. For similar reasons, target maturity levels that were defined in the Initial Validation Results [9] may be updated up until the end of the project to give a retrospective view on what has been achieved.

1.3 Report structure

This document is built on the test cases, key performance indicators and maturity levels defined as part of the Validation Framework [3]. It follows the structure of the original document, adding the results of the test cases, the values of the key performance indicators and the determined maturity levels as necessary. Since the methodology was lightly updated in the Initial Validation Results [9] by incorporating TRLs, this document continuous with this simultaneous mapping of maturity levels and technology readiness levels.

In cases where requirements have changed or test cases needed to be adjusted, it documents the change and evaluates the updated version.

Chapter 2 is the main section of this document: it lists the results of evaluating KPIs and test cases for all of the eight main components [6] of the CPSwarm platform and for the User Experience as a ninth component.

Chapter 3 summarizes the findings by calculating the maturity level of each component – as well as the project as a whole – based on the results.

Chapter 4 describes use case evaluation by drawing up a matrix between the requirements and their relevance to the three CPSwarm vision scenarios: SAR, Platoon and Logistics.

Chapter 5 summarizes results of the validation and evaluation process throughout the 3 phases of the project and the final validation results.

1.4 Related documents

ID	Title	Reference	Version	Date
D2.3	Initial Requirements Report	[1]	1.0	2017-07-04
D2.7	Final Lessons Learned and Requirements Report	[2]	1.0	2019-02-14
D2.8	Validation Framework Specification	[3]	1.0	2018-06-29
D3.1	Initial System Architecture and Design Specification	[4]	1.0	2017-08-18
D3.3	Final System Architecture Analysis and Design Specification	[5]	1.0	2019-07-10

D3.6	Final CPSwarm Workbench and associated tools	[6]	1.0	2019-11-30
D6.5	Initial Integration of External Simulators	[7]	1.0	2018-06-30
D7.2	Final CPSwarm Abstraction Library	[8]	1.0	2019-12-20
D8.7	Initial Validation Results	[9]	1.0	2019-09-17

2 KPIs and test cases for the CPSwarm Workbench components

This chapter collects the results of the evaluation of test cases and KPIs defined in the Validation Framework (Section 3 of [3]) for each component of the CPSwarm Workbench. The structure of each sub-chapter is as follows:

- Description of formal or informal test cases (if any).
- Description of the identified KPI and associated maturity levels (if any).
- List of requirement IDs (CRD-<ID>) and the result of their evaluation – the requirements themselves are defined in Section 4.2 of [2].

Each requirement can have one of three states:

- Validated: requirement is met and covered by test cases.
- Partially validated: requirement is partially met and covered by test cases.
- Not validated: requirement is not met or not covered by test cases.

And one more state:

- Not required: the requirement was dropped due to a design decision made after the requirement was created.

The maturity levels (MLs) are defined in Section 4.3 of [2]. They are updated here in more clarity. Maturity Levels are ideally applicable in software development, however, the project also develops technology, therefore the Technology Readiness Levels [10] (TRLs), as industry-standard measure are also included and matched with MLs according to this table:

<table>
<tr><th colspan="2">Maturity Levels</th><th colspan="2">Technology Readiness Levels</th></tr>
<tr><td>ML1</td><td>Proof of concept</td><td>TRL3</td><td>Experimental proof of concept</td></tr>
<tr><td>ML2</td><td>Core features implemented (basic core features are available, basic documentation is available)</td><td rowspan="2">TRL4</td><td rowspan="2">Technology validated in lab</td></tr>
<tr><td>ML3</td><td>Complete solution validated in lab environment (all the planned features, including the more advanced ones, are available, good code quality, good stability, complete documentation is available)</td></tr>
<tr><td>ML4</td><td>Additional features implemented and solution optimized to support operations in relevant environment. Documentation is available</td><td rowspan="2">TRL5</td><td rowspan="2">Technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies</td></tr>
<tr><td>ML5</td><td>Enhanced Solution tested and validated in relevant environment</td></tr>
</table>

Each test case can verify one or more requirements. A requirement is considered validated when all the test cases referencing it give a 'Passed' result. To determine the maturity level of a particular component, relevant

KPIs are compared to thresholds set for each of the five maturity levels defined above – the highest one below the KPI value is considered to be the maturity level for the component.

Note that some requirements represent high-level concerns that go beyond the component in question – while the tests are done within the context of the component and the results of the evaluation are discussed separately. In this evaluation there is one such requirement set, 'User Experience'; all requirements related to the UX of the CPSwarm platform are displayed in Section 2.9.

For some of the requirements the associated maturity level had to be updated, as at the start of the project, when the requirements were defined, the complexity of each task could not be foreseen correctly. In most cases, integration-related and more complex behaviours had to be changed to have higher maturity levels to reflect the complexity of the effort necessary to meet and implement such requirements.

The following requirements have no functional equivalents and as such will not be verified by any of the test cases defined:

List of Requirements for User Experience
CRD-38 The swarm should consist of self-organizing swarm members
CRD-49 All the swarm members of a swarm should act under only one mission at a time
CRD-50 The Mission Planner should be able to add constraints to a mission

The following sections go through the list of requirements for the CPSwarm Workbench components [6].

2.1 Modelling Tool

2.1.1 Evaluation

Metric name	The Modelling Tool is able to use / reuse models from the Modelling Library	
Verified requirements	CRD-2, CRD-30	
Maturity level	ML3, TRL4	
Steps to perform	Open the Modelling Tool, create a CPSwarm project and then open the Modelling Library option. Drag and drop models contained in the Modelling Library to the actual project created.	
Expected results	The loaded models can be used as they are or can be tailored according to specific needs (see the following test cases).	
Use-case relevance	All use-cases models reuse *Hardware functions* which are part of the Modelling Library.	
Results	The Modelling Library is loaded by default and can be used.	Passed

Metric name	The Modelling Tool shall be able to model the structure of a swarm member	
Verified requirements	CRD-3, CRD-21, CRD-33	
Maturity level	ML3, TRL4	
Steps to perform	Open a CPSwarm project or create a new one in the Modelling Tool. Assemble the model of the target swarm member using the Swarm Member Architecture diagram palette: add actuators, controllers, sensors, data flow indicators etc. to the model to represent the internal architecture of the swarm member.	
Expected results	The created diagram represents the structure of the swarm member.	
Use-case relevance	Both the Logistics and SAR use-cases have specified a swarm member, e.g. scout or drone architecture.	
Results	A swarm member architecture diagram has been implemented.	Passed

Metric name	The Modelling Tool shall be able to model the behaviour of a swarm member	
Verified requirements	CRD-4, CRD-32	
Maturity level	ML5, TRL5	
Steps to perform	Open a CPSwarm project or create a new one in the Modelling Tool. Assemble the behavioural model of the target swarm member using the Behavioural Modelling diagram palette: add states, transitions and pseudo-states to the model to represent the behaviour of the swarm member.	
Expected results	The created diagram represents the behaviour of the swarm member as a state machine.	
Use-case relevance	Behaviours have been model for all three use cases.	
Results	State machine diagrams for behaviour modelling are available.	Passed

Metric name	The Modelling Tool shall be able to model the composition of a swarm	
Verified requirements	CRD-6, CRD-48	
Maturity level	ML3, TRL4	
Steps to perform	Open a CPSwarm project or create a new one in the Modelling Tool. Assemble the model of the target swarm using the Swarm Architecture diagram palette: add swarms, swarm members, interfaces, attributes etc. to the model to represent the architecture of the swarm.	
Expected results	The resulting diagram can represent the composition of the swarm.	
Use-case relevance	A swarm architecture has been created for both the Logistic and SAR use-cases.	
Results	A Swarm architecture diagram has been defined and implemented.	Passed

Metric name	The Modelling Tool shall be able to model fitness function to define the goal of the swarm behaviour	
Verified requirements	CRD-7, CRD-31	
Maturity level	ML5, TRL5	
Steps to perform	Open a CPSwarm project or create a new one in the Modelling Tool. Establish the model of the fitness function using the Fitness Function Specification diagram palette: add the Fitness Function, parts representing the internal instantiations of components, ports for data flow communication between components, attributes etc. to the model to represent the fitness function.	
Expected results	The fitness function can be passed to the Optimization Tool and the optimization can be generated (see CRD-12, CRD-20).	
Use-case relevance	Both Logistic and SAR use-cases includes fitness function diagram.	
Results	A fitness function diagram has been defined, implemented and released.	Passed

Metric name	The Modelling Tool is responsible for passing swarm member structure to the Code Generator	
Verified requirements	CRD-54, CRD-55	
Maturity level	ML4, TRL5	
Steps to perform	When the project to be exported is ready, choose the option in the Modelling Tool which generates the definition of the swarm member structure and behaviour in a standardized form.	
Expected results	The files generated by the Modelling Tool can be used as valid inputs to the Code Generator.	
Use-case relevance	SCXML export has been used by the Code Generator in both the Logistic and SAR use-cases.	
Results	SCXML export is available under the Modelling tool and can be used by the Code Generator.	Passed

Metric name	The Modelling Tool makes it possible to define events	
Verified requirements	CRD-62, CRD-65, CRD-66, CRD-69, CRD-100, CRD-101	
Maturity level	ML3, TRL4	
Steps to perform	When creating a low-level state machine for describing the behaviour of a swarm member, create event trigger points connected to e.g. input/output values and define events that can refer to other behaviours in the high-level state machine. Mark these events according to their scope – swarm, swarm member or component, where swarm and swarm member scope events have to be handled as privileged commands.	
Expected results	The high and low-level state machines that describe the behaviour of a swarm member accurately describe the input and output values that can trigger a change in behaviour in different scopes including components, swarm members or the whole swarm.	
Use-case relevance	The modelling of all three use-cases include behaviours and related events.	
Results	Event and related data modelling are available under the Modelling Tool	Passed

Metric name	The Modelling Tool makes it possible to design swarm members with multiple behaviours	
Verified requirements	CRD-77, CRD-87, CRD-47	
Maturity level	ML5, TRL5	
Steps to perform	Define low level state machines for the desired behaviours of to the swarm member. Start defining a high-level state machine and perform the steps described in the test "The Modelling Tool makes it possible to define events".	
Expected results	The high-level state machine now defines the logic and transition between the different behaviours.	
Use-case relevance	The modelling of all three use-cases include multi-level behaviours.	
Results	The Modelling tool allows multi-level behaviour to be defined including transitions between them.	Passed

Metric name	The Modelling Tool shall be able to model fitness function to define the goal of the swarm behaviour	
Verified requirements	CRD-7, CRD-31	
Maturity level	ML4, TRL5	
Steps to perform	Open a CPSwarm project or create a new one in the Modelling Tool. Establish the model of the fitness function using the Fitness Function Specification diagram palette: add the Fitness Function, parts representing the internal instantiations of components, ports for data flow communication between components, attributes etc. to the model to represent the fitness function.	
Expected results	The fitness function can be passed to the Optimization Tool and the optimization can be generated (see CRD-12, CRD-20).	
Use-case relevance	Both the Logistic and SAR use-cases include fitness function modelling.	
Results	A fitness function diagram has been defined, implemented and released.	Passed.

Metric name	The Optimization Tool is integrated with the Modelling Tool	
Verified requirements	CRD-9, CRD-10, CRD-11, CRD-12, CRD-13, CRD-31, CRD-117	
Maturity level	ML4, TRL5	
Steps to perform	The Modelling Tool has to pass the end condition of simulation, environment model, swarm model, fitness function and swarm composition to the Optimization Tool: Define a fitness function that describes the goal of the swarm using the Modelling Tool Generate an Optimization Project using the corresponding module in the Modelling Tool Export the current project's parameters to the Optimization Tool. The generated files containing the parameters defined in the Modelling Tool shall be saved in the dedicated folder from which the Optimization Tool can load and use them.	
Expected results	The optimization should be able to run using the parameters provided.	
Use-case relevance	In Logistic use-case, a set of parameters to be optimised has been selected and exported, from the Modelling Tool, to be used by the Optimisation Tool.	
Results	A set of parameters to be optimised can be passed to Optimisation tool.	Not required

Metric name	The Modelling tool shall be able to model communication between swarm members and the environment
Verified requirements	CRD-5, CRD-70, CRD-8
Maturity level	ML5, TRL5
Steps to perform	Open a CPSwarm project or create a new one in the Modelling Tool. Assemble the model of the target swarm using the Swarm Architecture diagram palette: add swarms, swarm members, interfaces, attributes etc. to the model to represent the architecture of the swarm.
Expected results	The resulting diagram can represent the communication between swarm members and represent the environment.

	The Modelling Tool supports communication interface models and the Code Generator uses that information to target the correct interface.	
Use-case relevance	Communication configuration between swarm member has been generated for both Logistic and SAR use-cases.	
Results	Communication configuration can be generated from the modelling tool.	Passed

Metric name	The Modelling Library shall include special behaviours to support the built-in behaviours of the CPS like emergency shutdown, flying home or refuelling.	
Verified requirements	CRD-82	
Maturity level	ML5, TRL5	
Steps to perform	Open a CPSwarm project or create a new one in the Modelling Tool. Assemble the model of the target swarm using the Swarm Architecture diagram palette: add swarms, swarm members, interfaces, attributes a contingency behaviours.	
Expected results	The resulting diagram can represent the built-in contingency behaviours.	
Use-case relevance	All three use-cases uses dedicated behaviours, included in the Modelling Library, to built-in contingency behaviours	
Results	Modelling Library deployed in the modelling tool provides such behaviours.	Passed

2.1.2 Requirements evaluation results

ID	Requirement	State
CRD-2	The Modelling Tool shall be able to use / reuse models from the Modelling Library	Validated
CRD-3	The Modelling Tool shall be able to model the structure of a swarm member	Validated
CRD-4	The Modelling Tool shall be able to model the behaviour of a swarm member	Validated
CRD-5	The Modelling Tool shall be able to model communication between swarm members	Validated

CRD-6	The Modelling Tool shall be able to model the composition of a swarm	Validated
CRD-7	The Modelling Tool shall be able to model fitness function to define the goal of the swarm behaviour	Validated
CRD-8	The Modelling tool shall be able to model the environment	Validated
CRD-9	The Modelling Tool shall pass the end condition of simulation to the Optimization Tool	Not required
CRD-10	The Modelling Tool shall pass the environment model to the Optimization Tool	Not required
CRD-11	The Modelling Tool shall pass the swarm model to the Optimization Tool	Not required
CRD-12	The Modelling Tool shall pass fitness function to the Optimization Tool	Not required
CRD-13	The Modelling Tool shall pass the swarm composition to the Optimization Tool	Not required
CRD-21	The Modelling Tool should be able to present the structural diagram of a swarm member	Validated
CRD-30	The Modelling Tool shall enable users to create models and publish them in a private library	Validated
CRD-31	The Modelling Tool shall contain an editor to formulate the fitness function	Validated
CRD-32	The Modelling Tool shall be able to model the behaviour of the swarm member using the swarm member behaviour library	Validated
CRD-33	The Modelling Tool shall be able to model a local state as a part of the swarm member structure	Validated
CRD-54	The Modelling Tool shall be responsible for passing swarm member structure to the Code Generator	Validated
CRD-55	The Modelling Tool shall be responsible for passing swarm member behaviour to the Code Generator	Validated
CRD-62	The Modelling Tool shall make it possible to define events	Validated
CRD-65	The Modelling Tool shall distinguish between swarm, member and component scope events, which are defined at their respective level in the model hierarchy	Validated

CRD-66	The Modelling Tool shall make it possible to trigger events based on the current value of the inputs and outputs defined for the low-level behaviour of the current state	Validated
CRD-69	The Modelling Tool shall make it possible to add additional swarm scope events to each state transition that are triggered when the transition happens	Validated
CRD-70	The physical interface used for all communications and its parameters shall be configurable using the Modelling Tool.	Validated
CRD-77	The Modelling Tool shall make it possible to design systems with multiple behaviours where events can trigger a behaviour change	Validated
CRD-82	The Modelling Library shall include special behaviours to support the built-in behaviours of the CPS like emergency shutdown, flying home or refuelling.	Validated
CRD-87	The Modelling Tool shall let multiple high-level behaviours coexist within the same project	Validated
CRD-100	The Modelling Tool shall make it possible to specify event scope.	Validated
CRD-101	The Modelling Tool shall namespace component scope events to their respective component	Validated

2.2 Modelling Library

2.2.1 Evaluation

Metric name	The Swarm member library contains models for sensors and actuators to be used to design a swarm member				
Verified requirements	CRD-34, CRD-25, CRD-22, CRD-1				
Measurement	Count the number of models for the sensors and actuators to be used to design a swarm member in the Modelling Library. Only include completed models which have successfully been used in an example/vision scenario. From ML3 the Modelling Library should include use-case specific solutions for sensor capabilities.				
Target values	ML1	ML2	ML3	ML4	ML5
	3	5	8	10	15
Current value	17				

Use-case relevance	Both the Logistic and SAR use-cases use sensor and actuator models from the Modelling Library

Metric name	The swarm member library contains models for the physical aspects of the swarm member				
Verified requirements	CRD-74, CRD-25, CRD-22, CRD-1				
Measurement	Count the number of models for the physical aspects (e.g. sensors, controllers) of the swarm member in the Modelling Library. Only include completed, working models. Each of these shall possess component-scope events attached, for example events determined by input/output values.				
Target values	ML1	ML2	ML3	ML4	ML5
	2	4	8	12	15
Current value	20				
Use-case relevance	Both the Logistic and SAR uses model from the Modelling Library for swarm member physical aspect				

Metric name	The Swarm Member Library contains models for the behaviour of a swarm member				
Verified requirements	CRD-86, CRD-84, CRD-26, CRD-22, CRD-1				
Measurement	Count the number of models for the behaviour of a swarm member in the Modelling Library. Only include completed, working models. The minimum viable behaviour for ML2 is including the emergency exit example (or another toy-example), and from ML3 the Modelling Library should include behaviours connected to each of the use cases. ML4-5 should contain scenario and capability-specific contingency behaviours of a swarm member, including "Emergency stop shutdown" behaviours specific to the hardware platform used and a behaviour that describes the transition to manual remote control.				
Target values	ML1	ML2	ML3	ML4	ML5
	1	5	8 From ML3 including at least 1 soft shutdown	12 Including soft shutdown behaviours	18 Including a transitioning behaviour to manual

			contingency behaviour	for all hardware target platforms	remote control
Current value	21				
Use-case relevance	All three uses cases use the Modelling Library for behaviour modelling including specific contingency behaviours.				

Metric name	The environment library shall contain models of environments				
Verified requirements	CRD-28, CRD-24, CRD-23, CRD-1, CRD-27, CRD-71				
Measurement	Count the number of models for environments in the Modelling Library. Only include completed, working models.				
Target values	ML1	ML2	ML3	ML4	ML5
	0	1	3	4	5
Current value	2				
Use-case relevance	One environment has been modelled for the Logistic and SAR use-cases, but the number of models can be easily enriched by using the sdf import functionality provided by the Modelling Tool.				

Metric name	Number of different fitness functions related to different problems				
Verified requirements	CRD-29				
Measurement	Count the number of fitness functions related to different problems in the Modelling Library. Only include completed, working examples.				
Target values	ML1	ML2	ML3	ML4	ML5
	0	1 problem	1 problem	>1 problems	>1 problems

		1 fitness function	1 fitness function	at least 1 fitness function for each	at least 1 fitness function for each
Current value	2				
Use-case relevance	A fitness function has been defined for both the Logistic and SAR uses cases				

2.2.2 Requirements evaluation results

ID	Requirement	State
CRD-1	The Modelling Library will be a collection of different kinds of reusable components	Validated
CRD-22	The Modelling Library shall include a library to help in designing a swarm member	Validated
CRD-23	The Modelling Library shall include a library to help in designing an environment	Validated
CRD-24	The Modelling Library shall include a library to help in designing a goal	Validated
CRD-25	The swarm member library shall contain models for the physical aspects of the swarm member	Validated
CRD-26	The swarm member library shall contain models for the behaviour of a swarm member	Validated
CRD-27	The swarm member library shall contain models for the communication among swarm members	Validated
CRD-28	The environment library shall contain models of environments	Partially validated
CRD-29	The goal library shall contain various fitness functions linked to different problems	Validated
CRD-34	The Swarm member library shall contain models for sensors and actuators to be used to design a swarm member	Validated
CRD-71	The Modelling Library shall include CPS components for communication interfaces, with support for at least one type of mesh network.	Validated
CRD-74	Components in the Modelling Library can have component scope events associated with them, which are imported when the component is added	Validated

CRD-84	The Modelling Library shall include behaviours specific to target hardware platforms that can be used as safe default contingency plans for each CPS model (soft shutdown)	Validated
CRD-86	The Modelling Library shall include a special behaviour that switches over the CPS to manual remote control	Validated

2.3 Optimization Tool

2.3.1 Evaluation

Metric name	The Optimization Tool passes operational commands to the Optimization Simulator	
Verified requirements	CRD-14	
Maturity level	ML3, TRL4	
Steps to perform	Start the optimization using the Optimization Tool and the Optimization Simulator together.	
Expected results	The simulation can be performed, and the simulated swarm members behave as indicated by the Optimization Tool.	
Use-case relevance	Used to produce optimal swarm behaviour in the Logistics use-case.	
Results	The Optimization Tool communicates with the Simulation Tool using the shared XMPP communication library.	Passed

Metric name	The Optimization Tool shall optimize the algorithm according to the fitness score
Verified requirements	CRD-20, CRD-91
Maturity level	ML2, TRL4
Steps to perform	Create a fitness function that defines the goal of the swarm behaviour Start the optimization with the fitness function and other parameters that describe the swarm, including other behaviours to be included in the simulation, e.g. malicious behaviour of some agents, hardware failure, etc.

Expected results	The Optimization Tool is able to rank the candidate controllers according to the fitness score, and the optimization stops when the maximum of the fitness function is reached.	
Use-case relevance	Used to produce optimal swarm behaviour in the Logistics use-case.	
Results	The Optimization Tool instructs the Simulation Tool to perform a range of simulations and, after awaiting results, ranks candidate controllers by their fitness scores. Optimization is terminated early if the maximum fitness is reached.	Passed

Metric name	The Optimization Tool shall pass the optimal behaviour to the Code Generator	
Verified requirements	CRD-56, CRD-117	
Maturity level	ML2, TRL4	
Steps to perform	After the optimization is done, export the state machine that describes the optimized behaviour and load the file with the Code Generator.	
Expected results	The Code Generator can generate target platform specific code that implements the optimized behaviour.	
Use-case relevance	-	
Results	The Optimization Tool is not communicating with the Code Generator.	Not validated

2.3.2 Requirements evaluation results

ID	Requirement	State
CRD-14	The Optimization Tool shall pass operational commands to the Optimization Simulator	Validated
CRD-20	The Optimization Tool shall optimize the algorithm according to the fitness score	Validated
CRD-56	The Optimization Tool shall pass the optimal behaviour to the Code Generator	Not validated

CRD-91	The Optimization Tool shall only optimize one behaviour at a time, but shall let the simulation used include other behaviours	Validated

2.4 Simulation Tool

2.4.1 Evaluation

Metric name	The Optimization Simulator enables simulations that describe realistic scenarios.	
Verified requirements	CRD-15, CRD-16, CRD-19, CRD-88, CRD-90, CRD-42	
Maturity level	ML4, TRL5	
Steps to perform	Define the simulation using the Simulation Manager: Composition of the swarm – number of members, list of behaviours they can perform. Structure of swarm members – capabilities, hardware components. The model of the environment used for the simulation. Stochastic description of the occurrence of hardware faults. Start the simulation using the Optimization Simulator.	
Expected results	The Optimization Simulator simulates the scenario described by the environment model, swarm composition, behaviours and malicious events such as hardware faults. The Optimization Simulator can feed back the simulated input data collected by the swarm members into the Optimization Tool.	
Use-case relevance	The Simulation and Optimization Environment is integrated with the Workbench and it can be used to evaluate the performances of the swarm modelled with the Modelling Tool for both the SAR and Logistics scenario, using high-fidelity Simulation Tools.	
Results	The XMPP API defined in (The CPSwarm Project) allows the Optimization Tool and the Optimization Simulators to exchange the data and models to be used for the simulation.	Passed

Metric name	The Optimization Simulator creates and passes the fitness score to the Optimization Tool.
Verified requirements	CRD-17, CRD-18
Maturity level	ML4, TRL5

Steps to perform	Start the optimization using the Optimization Tool and the Optimization Simulator together.	
Expected results	After each of the iterations that simulate the behaviour generated by the Optimization Tool, the Optimization Simulator calculates a fitness score describing it and passes it to the Optimization Tool.	
Use-case relevance	The behaviour of the Swarm Logistics scenario is optimized using an evolutionary algorithm, guided by a fitness function modelled in the Modelling Tool and passed to the Simulation and Optimization Environment, in order to be used to calculate a fitness score for each set of parameters simulated. This fitness score is used to rank the candidate set of parameters and to select the best ones.	
Results	As expected, after having received the behaviour to be simulated from the Optimization Tool, the Optimization Simulator simulates it and calculates the fitness score using the defined fitness function, sending it back to the Optimization Tool.	Passed

2.4.2 Requirements evaluation results

ID	Requirement	State
CRD-15	The Optimization Simulator shall simulate swarm composition, swarm member structure	Validated
CRD-16	The Optimization Simulator shall simulate environment model	Validated
CRD-17	The Optimization Simulator shall calculate fitness score for each simulation	Validated
CRD-18	The Optimization Simulator shall pass the fitness score to the Optimization tool	Validated
CRD-19	The Optimization Simulator shall pass the sensor data of each swarm member back to the Optimization Tool	Not required
CRD-88	The Simulation Manager shall support simulations where different swarm members have different behaviours	Validated
CRD-90	The Simulation Manager shall support simulations where different hardware components are faulty or where faults occur stochastically	Validated

2.5 Code Generation Tool

2.5.1 Evaluation

Metric name	The Code Generator shall generate code for a multi-level state machine incorporating inputs from the Modelling Tool, the Optimization Tool and the user	
Verified requirements	CRD-94, CRD-97, CRD-102, CRD-117	
Maturity level	ML3, TRL4	
Steps to perform	Set up a project where some of the states in the behaviour are from the Modelling Library, others are generated by the Optimization Tool and others are stubbed out and left for the user to implement Have the Optimization Tool generate its own code, then implement the stubbed-out states in order to produce valid code the Code Generator can integrate Run the Code Generator	
Expected results	The Code Generator should generate code that is a valid state machine and can call the implementations supplied by the Optimization Tool and the user	
Notes	Relies on other workbench components to build the behaviour.	
Use-case relevance	For SAR and Logistic scenario was used to translate the Finite State Machine modelled algorithm (provided as input in form of SCXML file) into actual executable ROS code.	
Results	The Code Generator is able to generate code from state machines, but is not able to directly integrate code produced by the Optimization Tool.	Partially validated

Metric name	The Code Generator can target multiple platforms
Verified requirements	CRD-96
Maturity level	ML4, TRL5

Steps to perform	Perform the steps in the test "The Code Generator shall generate code for a multi-level state machine incorporating inputs from the Modelling Tool, the Optimization Tool and the user" for at least two different hardware platforms	
Expected results	For both platforms, the generated code is valid and can be deployed.	
Use-case relevance	Generated code was deployed on all platforms (drones and turtlebot/rovers) used in the SAR and Logistic scenarios	
Results	The Code Generator is able to target both a rover and a drone.	Passed

Metric name	The Code Generator shall generate code for a specific target platform	
Verified requirements	CRD-57	
Maturity level	ML4, TRL5	
Steps to perform	Set up a project where some of the states in the behaviour are from the Modelling Library, others are generated by the Optimization Tool and others are stubbed out and left for the user to implement Have the Optimization Tool generate its own code, then implement the stubbed-out states in order to produce valid code the Code Generator can integrate Run the Code Generator	
Expected results	The Code Generator should generate code that is a valid on the specified target platform	
Notes	Relies on other workbench components to build the behaviour.	
Use-case relevance	For both Logistic and SAR scenario the generated code targeted ROS as the specific running environment platform.	
Results	The Code Generator is able to target both a rover and a drone.	Passed

Metric name	The Code Generator and all the code generated shall be compliant to ISO 26262.	
Verified requirements	CRD-64	
Maturity level	ML2, TRL4	
Steps to perform	Set up a project where some of the states in the behaviour are from the Modelling Library, others are generated by the Optimization Tool and others are stubbed out and left for the user to implement Have the Optimization Tool generate its own code, then implement the stubbed-out states in order to produce valid code the Code Generator can integrate Run the Code Generator	
Expected results	The Code Generator should output code that is compliant with best practices described in the standard ensures that it can be used in automotive use cases.	
Notes	Relies on other workbench components to build the behaviour.	
Use-case relevance	Platooning use-case.	
Results	Code generation is not enabled for the platooning use case.	Not required

2.5.2 Requirements evaluation results

ID	Requirement	State
CRD-57	The Code Generator shall generate code for a specific target platform	Validated
CRD-63	The Code Generator shall generate code that is readable and understandable by humans.	Validated
CRD-64	The Code Generator and all the code generated shall be compliant to ISO 26262.	Not required
CRD-94	The Code Generator shall receive the model of the high-level behaviour as a state machine, with additional information passed about each state to define the inputs and outputs of the low-level behaviour that is being executed while that state is active	Validated

CRD-96	The Code Generator shall be configured to produce code for a specific platform.	Validated
CRD-97	The Code Generator shall integrate low-level behaviour algorithms generated by the Optimization Tool	Not required
CRD-102	The Code Generator shall integrate low-level behaviour algorithms implemented manually	Validated

2.6 Deployment Tool

2.6.1 Evaluation

Metric name	The Deployment Tool can deploy a new behaviour on a swarm member	
Verified requirements	CRD-58, CRD-59, CRD-61, CRD-79, CRD-51	
Maturity level	ML4, TRL5	
Steps to perform	Start the Deployment Tool Wait until the tool indicates that it has completed the enumeration or at most 1 minute Select a swarm member Initiate the deployment of a behaviour package	
Expected results	The Deployment Tool shows the progress of the deployment process, which ends successfully. The new behaviour can be observed as active on the swarm member.	
Notes	Relies on other workbench components to build the behaviour.	
Use-case relevance	The Deployment Tool achieves the expected results on the selected platforms for the Logistic and SAR use-cases.	
Results	The API enables deployment on selected members and provides progress information	Passed

Metric name	Deployed software artefacts are signed, and their signatures are verified

Verified requirements	CRD-72, CRD-73, CRD-75, CRD-76, CRD-78	
Maturity level	ML3, TRL4	
Steps to perform	Positive test	Negative test
	1. Set up the trust relationship between the swarm members and the Deployment Tool 2. Perform a deployment as described in the test "The Deployment Tool can deploy a new behaviour on a swarm member" 3. While the swarm member is inactive, corrupt the signature of the software package 4. Start the swarm member	1. Break or do not set up the trust relationship between the swarm members and the Deployment Tool. 2. Perform a deployment as described in the test "The Deployment Tool can deploy a new behaviour on a swarm member"
Expected results	The deployment itself should be successful. When the swarm member is activated after the signature has been corrupted, it should refuse to start its behaviour and shut down immediately.	Deployment should fail.
Notes	For platforms requiring compilation on the device, the positive test should not test the effects of corrupted signatures, since no signature should be present on the final executable.	
Use-case relevance	The mechanism is expected to function on Linux operating system, commonly used for the Logistic and SAR use-cases.	
Results	The signing and verification will not be implemented as part of the Deployment Tool. The integrity of packages is protected to some extent using Linux filesystem permissions. Protection beyond that requires hardware solutions to first protect the keys used for verification. Moreover, the execution of the behaviour is often performed by other system applications (e.g. systemctl, roslaunch) which are beyond the control of the Deployment Tool.	Not required

Metric name	The Deployment Tool can compile code before deployment	
Verified requirements	CRD-104, CRD-105	
Maturity level	ML4, TRL5	
Steps to perform	Perform the steps of the test "The Deployment Tool can deploy a new behaviour on a swarm member" with a package and platform combination that requires cross-compilation.	
Expected results	The Deployment Tool should show the results of the compilation before deployment has begun.	
Notes	Relies on other workbench components to build the behaviour.	
Use-case relevance	The compiled code is compatible with Linux operating system and ARM architecture, as expected for SAR devices.	
Results	The Deployment Tool provides an interface for the user to configure and perform cross-compilation.	Passed

Metric name	The Deployment Tool can compile code after deployment
Verified requirements	CRD-103, CRD-105
Maturity level	ML4, TRL5
Steps to perform	Perform the steps of the test "The Deployment Tool can deploy a new behaviour on a swarm member" with a package and platform combination that requires on device compilation.
Expected results	The Deployment Tool should show the results of the compilation after deployment has begun.
Notes	Relies on other workbench components to build the behaviour.

Use-case relevance	The Deployment Tool performs builds natively on linux/amd64 platforms as used by the Logistic use-case devices.	
Results	The API enables compilation on swarm members and reports the results.	Passed

Metric name	The Deployment Tool and the Deployment Agent communicate over a secure channel	
Verified requirements	CRD-60, CRD-73	
Maturity level	ML4, TRL5	
Steps to perform	Start capturing swarm communications Perform the steps of the test "The Deployment Tool can deploy a new behaviour on a swarm member" Stop capturing swarm communications Analyze the captured packets	
Expected results	The captured exchange meets state of the art cryptographic requirements.	
Notes	This test is not as exact as most other tests. Analysis should focus on ensuring that no parts of the deployment package are transmitted without encryption and that all the necessary authentication handshakes take place. The test should be repeated at various stages of the established trust relationship to see if authentication fails if it is required to fail.	
Use-case relevance	The Deployment Tool provides secure communication channels between the server and devices used in the Logistic and SAR scenarios.	
Results	All communication between the Deployment Agents are secured with asymmetric key encryption.	Passed

2.6.2 Requirements evaluation results

ID	Requirement	State
CRD-58	The Deployment Tool shall deploy artefacts on swarm members	Validated
CRD-59	The Deployment Agent shall report the deployment status	Validated

CRD-60	The communication between the Deployment Agent running on swarm members and the Deployment Manager shall be authenticated, authorized, encrypted, and integrity checked	Validated
CRD-61	The Deployment Manager shall receive the configuration of the deployment task from the operator prior to deployment	Validated
CRD-72	The Deployment Manager shall sign all packages with an operator specific key	Not required
CRD-73	The Deployment Tool shall implement secure over-the-air update functionality.	Validated
CRD-75	The Deployment Agent shall verify the signatures of packages on boot and when updates are received	Not required
CRD-76	The Deployment Manager shall provide a way to generate, import and export operator specific keys for code signatures	Validated
CRD-78	The Deployment Agent shall use the list of trusted certificates supplied when the device is first provisioned to validate signatures	Not required
CRD-79	The Deployment Agent shall be responsible for starting, stopping and monitoring the code that has been deployed, even during startups and shutdowns	Validated
CRD-103	The Deployment Tool shall provide the means to compile code on target platforms	Validated
CRD-104	The Deployment Tool shall provide the means to cross-compile code for the target platforms	Validated
CRD-105	The Deployment Tool shall provide the means to compile code	Validated

2.7 Abstraction Layer

2.7.1 Evaluation

Metric name	Remote soft shutdown requests are handled by the Abstraction Layer if the behaviour has no handler for them
Verified requirements	CRD-83
Maturity level	ML4, TRL5
Steps to perform	1. Set up a swarm where members are running a behaviour with no soft shutdown request handler 2. Start the Monitoring and Configuration Tool

	3. Wait until the tool indicates that it has completed the enumeration or at most 1 minute 4. Issue a remote soft shutdown request to a swarm member	
Expected results	The swarm member shuts down safely.	
Notes	Relies on other workbench components to set up the swarm and issue the request.	
Use-case relevance	The drones and rovers of the SAR and Logistics use-cases are able to receive remote shutdown messages triggered by the Monitoring Tool.	
Results	A remote shutdown request can be sent by the Monitoring Tool	Passed

Metric name	Remote soft shutdown requests are passed to the behaviour by the Abstraction Layer
Verified requirements	CRD-83
Maturity level	ML4, TRL5
Steps to perform	1. Set up a swarm where members are running a behaviour which handles soft shutdown request in a distinctive manner 2. Start the Monitoring and Configuration Tool 3. Wait until the tool indicates that it has completed the enumeration or at most 1 minute 4. Issue a remote soft shutdown request to a swarm member
Expected results	The swarm member shuts down safely, in a manner consistent with the behaviour specified.
Notes	Relies on other workbench components to set up the swarm and issue the request.
Use-case relevance	The drones and rovers of the SAR and Logistics use-cases are able to receive remote shutdown messages triggered by the Monitoring Tool, which is propagated to the upper layers by the Communication Library.

Results	A soft shutdown request is handled by the Abstraction Layer.	Passed

Metric name	Remote hard shutdown requests are handled by the Abstraction Layer	
Verified requirements	CRD-83	
Maturity level	ML2, TRL4	
Steps to perform	1. Start the Monitoring and Configuration Tool 2. Wait until the tool indicates that it has completed the enumeration or at most 1 minute 3. Issue a remote hard shutdown request to a swarm member	
Expected results	The swarm member shuts down safely.	
Notes	Relies on other workbench components to set up the swarm and issue the request.	
Results	A hard shutdown request cannot be managed by the Abstraction Layer	Not required

Metric name	If the behaviour is unresponsive, the Abstraction Layer translates the soft shutdown request into a hard shutdown request
Verified requirements	CRD-83, CRD-85
Maturity level	ML4, TRL5
Steps to perform	1. Set up a swarm where members are running a purposefully unresponsive behaviour 2. Start the Monitoring and Configuration Tool 3. Wait until the tool indicates that it has completed the enumeration or at most 1 minute 4. Issue a remote soft shutdown request to a swarm member
Expected results	The swarm member shuts down safely, in a manner consistent with how hard shutdown requests should be handled.

Notes	Relies on other workbench components to set up the swarm and issue the request.	
Results	This functionality is still not provided by the Abstraction Layer	Not required

Metric name	Number of sensors and actuators supported by the Abstraction Library				
Verified requirements	CRD-98, CRD-118				
Measurement	Sensors and actuators should be visible in the Modelling Library as building blocks. Count the number of building blocks that reference sensors and actuators.				
Target values	ML1	ML2	ML3	ML4	ML5
	1	2	4	7	11
Current value	4				

Metric name	Number of high-level CPS routines supported by the Abstraction Library				
Verified requirements	CRD-99				
Measurement	High-level CPS routines should be visible in the Modelling Library as building blocks. Count the number of building blocks that reference behaviours implemented by the Abstraction Library.				
Target values	ML1	ML2	ML3	ML4	ML5
	1	2	3	4	5
Current value	6				

Metric name	The swarm members should be able to communicate with each other

Verified requirements	CRD-40, CRD-117	
Maturity level	ML3, TRL4	
Steps to perform	Use the CPSwarm Communication Library in the swarm firmware Publish telemetry data in member 1 Subscribe to telemetry data in member 2 Receive telemetry data in member 2	
Expected results	The swarm members are able to publish and subscribe to data.	
Notes	Relies on other workbench components to set up the swarm.	
Use-case relevance	All use cases employ the CPSwarm Communication Library as the main mode of communication.	
Results	All the swarm members should be able to pass information (sensory, statistical, positional etc.) to each other.	Passed

Metric name	All communications between the swarm and the tools in the workbench shall be authenticated, integrity protected and encrypted.
Verified requirements	CRD-67, CRD-68
Maturity level	ML5, TRL5
Steps to perform	Use the CPSwarm Communication Library in the swarm firmware with security features enabled.
Expected results	All communications between the swarm and the tools in the workbench must use industry standard encryption and signature schemes.
Notes	Relies on other workbench components to build the behaviour. Deployment and monitoring should only be possible after authentication and with proper authorization. Messages in transit

	should be treated as confidential and must be protected against tampering and eavesdropping. Communications between swarm members mostly include remote events. Certain safety critical events need no encryption, and in fact benefit from reduced latency.	
Use-case relevance	All use cases employ the CPSwarm Communication Library as the main mode of communication.	
Results	The secure version of the Communication Library provides encrypted, authenticated and integrity-protected messaging.	Passed

2.7.2 Requirements evaluation results

ID	Requirement	State
CRD-40	The swarm members should be able to communicate with each other	Validated
CRD-67	All communications between the swarm and the tools in the workbench shall be authenticated, integrity protected and encrypted.	Validated
CRD-68	All communications between swarm members shall be authenticated and integrity protected, with a per-message policy on encryption.	Validated
CRD-83	The Abstraction Layer shall have low level support for remote shutdown requests that work regardless of the status of the current behaviour	Partially validated
CRD-85	The Abstraction Layer shall implement a hardware specific safe remote shutdown behaviour that cannot be overridden by the current behaviour (hard shutdown)	Not required
CRD-98	The Abstraction Layer shall provide APIs to access/control/set-up sensors and actuator on CPSs	Partially validated
CRD-99	The Abstraction Layer shall provide primitives to activate and control high-level CPS routines	Validated

2.8 Monitoring Tool

2.8.1 Evaluation

Metric name	The Monitoring and Configuration Tool can enumerate the members of a swarm
Verified requirements	CRD-36, CRD-37, CRD-39, CRD-45, CRD-46, CRD-107

Maturity level	ML2, TRL4	
Steps to perform	1. Start the Monitoring and Configuration Tool 2. Wait until the tool indicates that it has completed the enumeration or at most 1 minute	
Expected results	The Monitoring and Configuration Tool shows all active swarm members.	
Notes	The Monitoring and Configuration Tool should be started on a system that has already established a connection with the swarm or on a system that is capable of establishing such a connection using the features built into the tool itself. The swarm should have at least one active member.	
Use-case relevance	Required in all use cases and implemented	
Results	The Monitoring Tool shows all active swarm members.	Passed

Metric name	The Monitoring and Configuration Tool can enumerate properties of a swarm member	
Verified requirements	CRD-36, CRD-37, CRD-39, CRD-45, CRD-46	
Maturity level	ML3, TRL4	
Steps to perform	1. Perform the steps as defined in the test "The Monitoring and Configuration Tool can enumerate the members of a swarm" 2. Query one of the swarm members for its properties	
Expected results	The Monitoring and Configuration Tool shows all properties of the swarm member, including the type of the property and whether it is read-only or writable.	
Use-case validation	Required in all use cases and implemented and validated by use of tool w.r.t. this metric.	
Results	The Monitoring Tool shows the properties of the swarm member.	Passed

Metric name	The Monitoring and Configuration Tool can issue commands to individual swarm members	
Verified requirements	CRD-89, CRD-41, CRD-43, CRD-44, CRD-45	
Maturity level	ML4, TRL5	
Steps to perform	1. Perform the steps as defined in the test "The Monitoring and Configuration Tool can enumerate the members of a swarm" 2. Issue a command to one of the swarm members	
Expected results	The swarm member reacts to the command and performs the associated action.	
Use-case validation	Required in all use cases and implemented and validated by use of tool w.r.t. this metric.	
Results	The Monitoring Tool issued the command and the swarm member reacted as expected.	Passed

Metric name	The Monitoring and Configuration Tool can enable the user to launch an external tool to take direct control of a swarm member	
Verified requirements	CRD-92, CRD-41, CRD-118	
Maturity level	ML4, TRL5	
Steps to perform	1. Perform the steps as defined in the test "The Monitoring and Configuration Tool can enumerate the members of a swarm" 2. Issue a request to take remote control of a swarm member 3. Wait until the response of approval and then launch the external tool	
Expected results	An external tool is launched and can be used to control the swarm member directly.	
Notes	Not all swarm members need to be compatible with this feature. Ensure that the selected swarm member has an associated external control tool and that handover is enabled on the device.	
Use-case relevance	Required in all use cases and implemented	
Results	Covered and V&V performed by checking for function and availability	Passed

Metric name	The Monitoring and Configuration Tool can observe events as they happen on swarm members	
Verified requirements	CRD-93, CRD-39, CRD-45, CRD-46, CRD-115	
Maturity level	ML5, TRL5	
Steps to perform	1. Perform the steps as defined in the test "The Monitoring and Configuration Tool can enumerate the members of a swarm" 2. Trigger an event on one of the swarm members manually	
Expected results	The Monitoring and Configuration Tool show the event as it happens.	
Notes	A special behaviour on the swarm member might be necessary to perform this test. The test should be repeated for each event scope to ensure that all event scopes are monitored correctly.	
Use-case relevance	Required in all use cases and implemented	
Results	Covered and V&V performed by checking for function and availability	Passed

Metric name	The communication link between the swarm and the Monitoring Tool shall be authenticated and encrypted
Verified requirements	CRD-35
Maturity level	ML5, TRL5
Steps to perform	Use the Secure Communication Library in the Monitoring Tool and swarm members
Expected results	The encryption and authentication should be SOA
Notes	Data received from swarm needs to stay confidential. The confidential data received from the swarm should not be accessed by an unauthorized entity. Protection of the swarm from attacks.
Use-case relevance	Required in all use cases and implemented

Results	Covered and V&V performed by checking for function and availability	Passed

Metric name	The Monitoring Tool shall allow visualization of the location of the swarm members	
Verified requirements	CRD-106	
Maturity level	ML4, TRL5	
Steps to perform	Open the Monitoring Tool and check the location of the added swarm members.	
Expected results	The swarm members are displayed on a map.	
Use-case relevance	Required in all use cases and implemented	
Results	Maps function implemented and works in all use cases	Passed

Metric name	The Monitoring Tool shall allow the selection of individual swarm members, and show availability	
Verified requirements	CRD-108, CRD-114	
Maturity level	ML3, TRL4	
Steps to perform	Open the Monitoring Tool and click on a swarm member from the list or the map.	
Expected results	The individual swarm member clicked is selected and additional data/commands are displayed related to the selected swarm member.	
Use-case relevance	Required in all use cases and implemented	
Results	Selection supported and verified in all use cases	Passed

Metric name	The Monitoring Tool shall be able to trigger events	
Verified requirements	CRD-109	
Maturity level	ML4, TRL5	
Steps to perform	Open the Monitoring Tool and trigger an event.	
Expected results	The swarm members are displayed on a map.	
Notes	Ability to manually trigger events in the swarm, either to selected swarm members or the whole swarm (using the discovery of the communication library).	
Use-case relevance	Required in all use cases and implemented	
Results	Covered and V&V performed by checking for function and availability	Passed

Metric name	The Monitoring Tool shall have hotkeys	
Verified requirements	CRD-110	
Maturity level	ML3, TRL4	
Steps to perform	Open the Monitoring Tool and select a swarm member. Press a key (e.g. s) to start an event.	
Expected results	Hotkeys are available for events such as: Start, Stop	
Use-case relevance	Required in all use cases and implemented	
Results	Covered and V&V performed by checking for function and availability	Passed

Metric name	The Monitoring Tool shall be able to set global parameters to the swarm

Verified requirements	CRD-111	
Maturity level	ML5, TRL5	
Steps to perform	Open the Monitoring Tool and select "Set global parameters" menu item.	
Expected results	The user is able to set global parameters to the swarm: distance_object (distance to keep from front object), distance_platoon (distance to keep a platoon member), cruise speed (average speed in platoon mode), max speed (hard limit on the speed)	
Use-case relevance	Required in all use cases and implemented	
Results	Covered and V&V performed by checking for function and availability	Passed

Metric name	The Monitoring Tool shall be able to show the cart assigned to each swarm member	
Verified requirements	CRD-112	
Maturity level	ML5, TRL5	
Steps to perform	Open the Monitoring Tool and check the visualization.	
Expected results	The user is able to see which cart is assigned to each robot.	
Notes	The functionality is associated with the Logistics scenario only.	
Use-case relevance	Required for the Logistics use case	
Results	Implemented and V&V performed by checking for the function in use case	Passed

Metric name	The Monitoring Tool shall be able to show the path of the swarm member

Verified requirements	CRD-113	
Maturity level	ML5, TRL5	
Steps to perform	Open the Monitoring Tool and check the map of the added swarm members.	
Expected results	The swarm members are displayed on a map showing the path that each robot has to follow.	
Notes	The functionality is associated with the Logistics scenario only.	
Use-case relevance	Required for the Logistics use case	
Results	Implemented and V&V performed by checking for the function in use case	Passed

2.8.2 Requirements evaluation results

ID	Requirement	State
CRD-35	The communication link between the swarm and the Monitoring Tool shall be authenticated and encrypted	Validated
CRD-36	The Modelling Tool shall provide the type of swarm member, type of data and data source to the Monitoring Tool	Validated
CRD-37	The Monitoring and Configuration Tool shall provide the type and address of swarm member	Validated
CRD-89	The Monitoring and Configuration Tool shall be able to trigger remote events on individual swarm members	Validated
CRD-92	The Monitoring and Configuration Tool shall enable the user to launch an external tool to take remote control of a specific swarm member	Validated
CRD-93	The Monitoring and Configuration Tool shall be able to monitor events in all scopes as they are being triggered by or received on a swarm member	Validated
CRD-106	The Monitoring Tool shall allow visualization of the location of the swarm members	Validated
CRD-107	The Monitoring Tool shall provide a list of swarm members	Validated

CRD-108	The Monitoring Tool shall allow the selection of individual swarm members	Validated
CRD-109	The Monitoring Tool shall be able to trigger events	Validated
CRD-110	The Monitoring Tool shall have hotkeys	Validated
CRD-111	The Monitoring Tool shall be able to set global parameters to the swarm	Validated
CRD-112	The Monitoring Tool shall be able to show the cart assigned to each swarm member	Validated
CRD-113	The Monitoring Tool shall be able to show the path of the swarm member	Validated
CRD114	The Monitoring Tool shall show how many swarm members are available	Validated

2.9 User Experience

2.9.1 Evaluation

High-level user experience requirements have been verified by the test cases defined for their relevant components, as follows:

- CRD-47, CRD-48, CRD-117: Modelling Tool (2.1)
- CRD-117: Optimization Tool (2.3)
- CRD-42: Simulation Tool (2.4)
- CRD-117: Code Generation Tool (2.5)
- CRD-51: Deployment Tool (2.6)
- CRD-117, CRD-118: Abstraction Layer (2.7)
- CRD-39, CRD-41, CRD-43, CRD-44, CRD-45, CRD-46, CRD-115, CRD-117, CRD-118: Monitoring Tool (2.8)

Metric name	Software components running on the CPS shall be started with the lowest possible privileges.
Verified requirements	CRD-81, CRD-119, CRD-123, CRD-126, CRD-127, CRD-128, CRD-143
Maturity level	ML3, TRL4
Steps to perform	Check user role settings for the executables available in ROS. GDPR is enforced by secure storage of user data. Perform penetration testing against the system.
Expected results	The software components should run as normal user without elevated privileges. User data should be protected and handled according to GDPR. Access to sensitive data is logged by user ID and timestamp. Unauthorized attempts are also logged. Firewalls and system are configured to resist to

	reasonable cyber-attacks. Unusual behaviour is resulting an alert to administrators. Passwords are never displayed.	
Results	Super user rights are required for UWB communication on the drones (SAR scenario). D7.2 lists ROS hardening methods to secure the operating system. Live processing of video data stays on device. Web access for Deployment Tool stores user database locally.	Passed

Metric name	System should provide guides and other material for training of users	
Verified requirements	CRD-116, CRD-135, CRD-136, CRD-137	
Maturity level	ML3, TRL4	
Steps to perform	The system is intuitive from the firs time of use. Check the Launcher for documentation.	
Expected results	The Launcher should provide documentation which can be used as guide and training material, which is sufficiently detailed to teach users to be experts. Novice users are able to use the Workbench without training to create simple swarm systems.	
Results	The following resources are available for learning and training: Wikis, Videos, Launcher How-tos, Examples in the Modelling Tool	Passed

Metric name	The Workbench components should be responsive scalable and support a large number of entities
Verified requirements	CRD-121, CRD-122, CRD-124, CRD-125
Maturity level	ML3, TRL4
Steps to perform	Add a large number of swarm members from modelling to deployment. Observe the performance of the tools with a large number of swarm members and users.
Expected results	The system is scalable to support massive growth in the number of users/devices, etc. All interfaces have reasonable response times.

	The system shall be evaluated by 85% of the professional users to be easy to use and generate no additional workload.	
Results	Limited tests showed general user satisfaction with the usage of tools.	Passed

Metric name	The system shall not use picture icons that could be considered offensive in any country where the system is used	
Verified requirements	CRD-129	
Maturity level	ML3, TRL4	
Steps to perform	Check all system GUI for offensive content.	
Expected results	The system contains no offensive content	
Results	To the best of our knowledge there are no offensive content in the developed tools.	Passed

Metric name	Updates are non-destructive and incremental	
Verified requirements	CRD-130, CRD-131, CRD-138, CRD-139, CRD-140	
Maturity level	ML3, TRL4	
Steps to perform	Install a new version of the system and observe the configurations.	
Expected results	Installing an upgrade does not modify existing configuration values. New version of the main system can be upgraded from any previous version. In case the update fails, the previous version is rolled back. When operating after a failure the user is informed the application is operating in a "safe mode" and all data is available for review without update.	
Results	The Modelling Tool and the Deployment Tool can be updated. The Deployment Tool updates are non-destructive	Passed

	in case there is no change in the schema of the decoupled database.	

Metric name	Localization is supported for all GUI components.	
Verified requirements	CRD-132, CRD-134	
Maturity level	ML3, TRL4	
Steps to perform	Check source code for displayed and localized content.	
Expected results	No piece of text that might be displayed to a user shall reside in source code. The structure of the data store shall be such that multi-lingual support shall not necessitate additional components or the need to replace current components, and the user shall be able to nominate their preferred language when entering their personal information.	
Results	Modelling Tool offers English and French localization. Other tools use English.	Passed

Metric name	The system has high availability	
Verified requirements	CRD-133, CRD-141, CRD-142,	
Maturity level	ML3, TRL4	
Steps to perform	Shut down parts of the system, try to launch disabled components and observe notifications.	
Expected results	The system shall not be shut down for maintenance more than once in a 24-hour period. The system shall prevent access to failed functions while providing access to all currently operational functions. Unless the system is non-operational, the system shall present a user with notification informing them that the system is unavailable.	
Results	The Launcher, the Modelling Tool and the Monitoring Tool is running on the individual machines of the user/developer.	Passed

	The Deployment Tool and Optimization Tool can be scaled by adding more servers.	

2.9.2 Requirements evaluation results

ID	Requirement	State
CRD-39	The Swarm Operator should be able to monitor the swarm	Partially validated
CRD-41	The Swarm Operator should be able to change the mission on the go	Partially validated
CRD-42	Environment conditions should be simulated	Validated
CRD-43	The Mission Planner should be able to configure a mission	Validated
CRD-44	The Mission Planner should be able to start a mission	Validated
CRD-45	The Mission Planner would like to have a UI to configure a mission	Partially validated
CRD-46	The Swarm Operator would like to have a UI to monitor the swarm in play	Partially validated
CRD-47	The swarm can have heterogeneous or a homogeneous composition	Validated
CRD-48	The Swarm Designer should be able to define the composition of the swarm	Validated
CRD-51	The Swarm Designer should be able to assign role to swarm member	Validated
CRD-81	Software components running on the CPS shall be started with the lowest possible privileges.	Validated
CRD-115	The system shall be able to show/visualise relevant information in an understandable manner	Validated
CRD-116	System should provide guides and other material for training of users	Validated
CRD-117	All the components of the system shall be well integrated	Validated
CRD-118	The system should be able to interface and interoperate with existing systems	Validated
CRD-119	Data processing and management must comply with relevant regulations	Validated
CRD-121	Any interface between the user and the platform must have a reasonable response time	Validated
CRD-122	The system shall be scalable to support massive growth in the number of users/devices, etc.	Validated

CRD-123	The solution should be in compliance with GDPR as well as national policies	Validated
CRD-124	The system shall be evaluated by 85% of the professional users to be easy to use	Validated
CRD-125	The system shall not generate additional workload for the professional users	Validated
CRD-126	Accessing sensitive data must be logged (User ID, Timestamp, etc.)	Validated
CRD-127	Attempts at accessing sensitive data by unauthorised users must be logged	Validated
CRD-128	The system shall be protected against cyber attacks	Validated
CRD-129	The system shall not use picture icons that could be considered offensive in any country where the system is used	Validated
CRD-130	Installing an upgrade shall not modify existing configuration values	Validated
CRD-131	When a new version of the main system is released, it shall be possible to upgrade to it from any previous version	Validated
CRD-132	No piece of text that might be displayed to a user shall reside in source code	Validated
CRD-133	The system shall not be shut down for maintenance more than once in a 24-hour period	Validated
CRD-134	Provisions shall be made for the future usage of multiple languages	Validated
CRD-135	The system shall be useable by users after nominal training	Validated
CRD-136	People with no training and no understanding of English shall be able to use the product	Validated
CRD-137	The product shall be self-explanatory and intuitive	Validated
CRD-138	When an update failure is detected all updates performed during the failed session shall be rolled back to restore the data to pre-session condition	Validated
CRD-139	All data recovered in a roll-back condition shall be recorded for use in forward recovery under user control	Validated
CRD-140	When operating after a failure the user shall be informed the application is operating in a “safe mode” and all data is available for review without update	Validated

CRD-141	The system shall prevent access to failed functions while providing access to all currently operational functions	Validated
CRD-142	Unless the system is non-operational, the system shall present a user with notification informing them that the system is unavailable	Validated
CRD-143	Passwords shall never be viewable at the point of entry or at any other time	Validated

3 Maturity

The previous tests result in a maturity level for the components based on averaging the test results. Positive results (green) are counted as their maturity level number, negative results (red) are counted as their maturity level minus one. The Final maturity level for the component is the down-rounded value of the average score. Dropped requirements are marked with yellow.

Components:	Modelling Tool	Modelling Library	Optimization Tool	Simulation Tool	Code Generation	Deployment Tool	Hardware Abstraction	Monitoring Tool
Final Maturity level for the component:	**ML4**	**ML4**	**ML2**	**ML4**	**ML4**	**ML4**	**ML4**	**ML4**
The Modelling Tool is able to use / reuse models from the Modelling Library	ML3							
The Modelling Tool shall be able to model the structure of a swarm member	ML3							
The Modelling Tool shall be able to model the behaviour of a swarm member	ML5							
The Modelling Tool shall be able to model the composition of a swarm	ML3							
The Modelling Tool shall be able to model fitness function to define the goal of the swarm behavior	ML5							
The Optimization Tool is integrated with the Modelling Tool	N/A							
The Modelling Tool is responsible for passing swarm member structure to the Code Generator	ML4							
The Modelling Tool makes it possible to define events	ML3							
The Modelling Tool makes it possible to design swarm members with multiple behaviors	ML5							
The Modelling Tool shall be able to model fitness function to define the goal of the swarm behaviour	ML4							
The Modelling Library shall include special behaviours to support the built-in behaviours of the CPS like emergency shutdown, flying home or refuelling.	ML4							
The Modelling Tool shall be able to model communication between swarm members and the environment	ML5							
The Swarm Member Library contains models for sensors and actuators to be used to design a swarm member		ML5						
The Swarm Member Library contains models for the physical aspects of the swarm member		ML5						
The Swarm Member Library contains models for the behaviour of a swarm member		ML5						

Components:	Modelling Tool	Modelling Library	Optimization Tool	Simulation Tool	Code Generation	Deployment Tool	Hardware Abstraction	Monitoring Tool
The Environment Library shall contain models of environments		ML2						
Number of different fitness functions related to different problems		ML5						
The Optimization Tool passes operational commands to the Optimization Simulator			ML3					
The Optimization Tool shall optimize the algorithm according to the fitness score			ML2					
The Optimization Tool shall pass the optimal behaviour to the Code Generator			N/A					
The Optimization Simulator enables simulations that describe realistic scenarios.				ML4				
The Optimization Simulator creates and passes the fitness score to the Optimization Tool.				ML4				
The Code Generator shall generate code for a multi-level state machine incorporating inputs from the Modelling Tool, the Optimization Tool and the user					N/A			
The Code Generator shall generate code for a specific target platform					ML4			
The Code Generator can target multiple platforms					ML4			
The Deployment Tool can deploy a new behaviour on a swarm member						ML4		
Deployed software artefacts are signed and their signatures are verified						N/A		
The Deployment Tool can compile code before deployment						ML4		
The Deployment Tool can compile code after deployment						ML4		
The Deployment Tool and the Deployment Agent communicate over a secure channel						ML4		
Remote soft shutdown requests are handled by the Abstraction Layer if the behaviour has no handler for them							ML4	
Remote soft shutdown requests are passed to the behaviour by the Abstraction Layer							ML4	
Remote hard shutdown requests are handled by the Abstraction Layer							N/A	
If the behavior is unresponsive, the Abstraction Layer translates the soft shutdown request into a hard shutdown request							N/A	

Components:	Modelling Tool	Modelling Library	Optimization Tool	Simulation Tool	Code Generation	Deployment Tool	Hardware Abstraction	Monitoring Tool
Number of sensors and actuators supported by the Abstraction Library							ML3	
Number of high-level CPS routines supported by the Abstraction Library							ML5	
The swarm members should be able to communicate with each other							ML3	
All communications between the swarm and the tools in the workbench shall be authenticated, integrity protected and encrypted.							ML5	
The Monitoring and Configuration Tool can enumerate the members of a swarm								ML2
The Monitoring and Configuration Tool can enumerate properties of a swarm member								ML3
The Monitoring and Configuration Tool can issue commands to individual swarm members								ML4
The Monitoring and Configuration Tool can enable the user to launch an external tool to take direct control of a swarm member								ML4
The Monitoring and Configuration Tool can observe events as they happen on swarm members								ML5
The communication link between the swarm and the Monitoring Tool shall be authenticated and encrypted								ML5
The Monitoring Tool shall allow visualization of the location of the swarm members								ML4
The Monitoring Tool shall provide a list of swarm members								ML4
The Monitoring Tool shall allow the selection of individual swarm members								ML4
The Monitoring Tool shall be able to trigger events								ML4
The Monitoring Tool shall have hotkeys								ML3
The Monitoring Tool shall be able to set global parameters to the swarm								ML5
The Monitoring Tool shall be able to show the cart assigned to each swarm member								ML5
The Monitoring Tool shall be able to show the path of the swarm member								ML5

4 Use case validation

The following table summarizes the relevance of the use cases for the requirements. Minus denotes that the requirement is not applicable for the use case. Plus, marks the cases where the high level requirement is relevant to the use case.

Use case relevance				Validation result
High-level requirement	**SAR**	**Platoon**	**Logistics**	
The Modelling Tool is able to use / reuse models from the Modelling Library	+	+	+	Validated
The Modelling Tool shall be able to model the structure of a swarm member	+	-	+	Validated
The Modelling Tool shall be able to model the behaviour of a swarm member	+	+	+	Validated
The Modelling Tool shall be able to model the composition of a swarm	+	-	+	Validated
The Modelling Tool shall be able to model fitness function to define the goal of the swarm behaviour	+	-	+	Validated
The Optimization Tool is integrated with the Modelling Tool	-	-	+	Validated
The Modelling Tool is responsible for passing swarm member structure to the Code Generator	+	-	+	Validated
The Modelling Tool makes it possible to define events	+	+	+	Validated
The Modelling Tool makes it possible to design swarm members with multiple behaviors	+	+	+	Validated
The Modelling Tool shall be able to model communication between swarm members and the environment	+	-	+	Validated
The Swarm Member Library contains models for sensors and actuators to be used to design a swarm member	+	-	+	Validated
The Swarm Member Library contains models for the physical aspects of the swarm member	+	-	+	Validated
The Swarm Member Library contains models for the behaviour of a swarm member	+	+	+	Validated
The Environment Library shall contain models of environments	+	-	+	Validated
Number of different fitness functions related to different problems	+	-	+	Validated

Use case relevance				Validation result
High-level requirement	SAR	Platoon	Logistics	
The Optimization Tool passes operational commands to the Optimization Simulator	-	-	+	Validated
The Optimization Tool shall optimize the algorithm according to the fitness score	-	-	+	Validated
The Optimization Tool shall pass the optimal behaviour to the Code Generator	-	-	+	N/A
The Optimization Simulator enables simulations that describe realistic scenarios.	+	-	+	Validated
The Optimization Simulator creates and passes the fitness score to the Optimization Tool.	-	-	+	Validated
The Code Generator shall generate code for a multi-level state machine incorporating inputs from the Modelling Tool, the Optimization Tool and the user	+	-	+	N/A
The code generated by the Code Generator is tidy and readable	-	-	-	Validated
The Code Generator shall generate code for a specific target platform	+	-	+	Validated
The Code Generator and all the code generated shall be compliant to ISO 26262.	-	+	-	N/A
The Code Generator can target multiple platforms	+	-	+	Validated
The Deployment Tool can deploy a new behaviour on a swarm member	+	-	+	Validated
Deployed software artefacts are signed and their signatures are verified	+	-	+	N/A
The Deployment Tool can compile code before deployment	+	-	-	Validated
The Deployment Tool can compile code after deployment	-	-	+	Validated
The Deployment Tool and the Deployment Agent communicate over a secure channel	+	-	+	Validated
Remote soft shutdown requests are handled by the Abstraction Layer if the behaviour has no handler for them	+	-	+	Validated
Remote soft shutdown requests are passed to the behaviour by the Abstraction Layer	+	-	+	Validated

Use case relevance				Validation result
High-level requirement	SAR	Platoon	Logistics	
Remote hard shutdown requests are handled by the Abstraction Layer	-	-	-	N/A
If the behaviour is unresponsive, the Abstraction Layer translates the soft shutdown request into a hard shutdown request	-	-	-	N/A
All communications between the swarm and the tools in the Workbench shall be authenticated, integrity protected and encrypted.	+	+	+	Validated
The Monitoring and Configuration Tool can enumerate the members of a swarm	+	+	+	Validated
The Monitoring and Configuration Tool can enumerate properties of a swarm member	+	+	+	Validated
The Monitoring and Configuration Tool can issue commands to individual swarm members	+	+	+	Validated
The Monitoring and Configuration Tool can enable the user to launch an external tool to take direct control of a swarm member	+	+	+	Validated
The communication link between the swarm and the Monitoring Tool shall be authenticated and encrypted	+	+	+	Validated
The Monitoring Tool shall allow visualization of the location of the swarm members	+	+	+	Validated
The Monitoring Tool shall provide a list of swarm members	+	+	+	Validated
The Monitoring Tool shall allow the selection of individual swarm members	+	+	+	Validated
The Monitoring Tool shall be able to trigger events	+	+	+	Validated
The Monitoring Tool shall have hotkeys	+	+	+	Validated
The Monitoring Tool shall be able to set global parameters to the swarm	+	+	+	Validated
The Monitoring Tool shall be able to show the cart assigned to each swarm member	-	-	+	Validated
The Monitoring Tool shall be able to show the path of the swarm member	+	+	+	Validated
The Monitoring Tool shall show how many swarm members are available	+	+	+	Validated

5 Summary

As defined in the project proposal, the CPSwarm project has three major phases – synchronized with the three years of the project. For the end of Phase 2 and Phase 3, a target maturity level was set for each component [3]. Based on the results of the test cases and the current value of the key performance indicators as described in the previous chapter, the table below summarizes the final maturity level and TRL of each component and the project as a whole.

		MS9 – Phase 2 (planned)		**MS9 – Phase 2 (actual)**		**MS12 – Phase 3 (planned)**		**MS12 – Phase 3 (actual)**	
Components	Modelling Tool	ML2	TRL4	ML1	TRL3	**ML4-5**	**TRL5**	**ML4**	**TRL5**
	Modelling Library	ML2	TRL4	ML1	TRL3	**ML4-5**	**TRL5**	**ML4**	**TRL5**
	Optimization Tool	ML2	TRL4	ML2	TRL4	**ML2-3**	**TRL4**	**ML2**	**TRL4**
	Simulation Tool	ML2	TRL4	ML3	TRL4	**ML4-5**	**TRL5**	**ML4**	**TRL5**
	Code Generation Tool	ML2	TRL4	ML1	TRL3	**ML4-5**	**TRL5**	**ML4**	**TRL5**
	Deployment Tool	ML2	TRL4	ML1	TRL3	**ML4-5**	**TRL5**	**ML4**	**TRL5**
	Hardware Abstraction Layer	ML2	TRL4	ML2	TRL4	**ML2-3**	**TRL5**	**ML4**	**TRL5**
	Monitoring Tool	ML2	TRL4	ML2	TRL4	**ML4-5**	**TRL5**	**ML4**	**TRL5**
Project		ML2	TRL4	ML1	TRL3	**ML2**	**TRL4**	**ML2**	**TRL4**

Going by strict rules of aggregating maturity levels (e.g. as suggested by Bolat[1]), the project's maturity is defined as that of the lowest of any of its subcomponents (ML2, TRL4 in this case). However, this does not accurately reflect the state of the project, as many of the project's goals can be accomplished by using only a subset of the tools that are already at least at Maturity Level 4, which is equivalent to TRL5.

During development of the project, the maturity of the components increased steadily. Some requirements that the Consortium approved at the planning phase of the projects were proven to be not necessary or optimal as the design decisions and evolvement of the implemented components were reflected back on the project and the original vision evolved. The requirements about integrating various components turned out to be the most effort-demanding in the project, as the dependencies were frequently a cause of delay, however, the Consortium managed to overcome these challenges and create the CPSwarm Workbench and its components as a coordinated and collaborative effort matching the target TRL levels.

[1] https://serkanbolat.com/2014/11/03/technology-readiness-level-trl-math-for-innovative-smes/

Using the Workbench developers and users of various level of expertise can create complex swarm-behaviour in a guided and systematic way. The clear advantages of using the tools created by the CPSwarm Consortium is that an integrated workbench can achieve more than just the individual parts without integration and interaction between the components.

Acronyms

Acronym	Explanation
KPI	Key Performance Indicator
ML	Maturity Level
CPS	Cyber-Physical System
OTA	Over-The-Air
TRL	Technology Readiness Level
UI	User Interface
UX	User eXperience

List of figures

References

[10] Horizon 2020 Work programme. (n.d.). *Technology readiness levels.* Retrieved from https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf