Project Proposal Summary Sheet

Project Number	**P_2022_03**
Domain	Simulation
Relevant Standard	ASAM OpenODD
Project Name	ASAM OpenODD V1.0.0
Project Type	☒Major ☐Minor ☐Revision ☐Concept ☐Implementation ☐Study
Start Date	01.08.2022
End Date	30.01.2024
ASAM Funds	150.000
Proposer	Siddartha Khastgir (WMG)

Table of Contents

1 Executive Summary 3

2 Overview / Goals 4

2.1 Motivation 4

2.2 User stories 5

2.3 Requirements 8

2.4 Relations to Other Standards, Projects or Organizations 10

3 Technical Content 11

3.1 Work Packages 11

3.1.1 WP1: Project coordination 11

3.1.2 WP2: Ramp-up and knowledge sharing 11

3.1.3 WP3: Scope Refinement 12

3.1.4 WP4: Language 13

3.1.5 WP5: Extensions 13

3.1.6 WP6: Case studies 13

3.2 Project Resources 13

3.3 Company Commitments 14

3.4 Budget 14

4 Review Steps 15

5 Deliverables 16

6 Project Plan 17

1 Executive Summary

Safety is fundamental to the development of public trust and acceptance of Connected and Autonomous Vehicles (CAVs) and their on board Automated Driving Systems (ADSs) and to enable deployment of automated driving. Safety of CAVs has two aspects: safe design and safe use of the system. In order to ensure safe use of the system, it is important to convey the knowledge of the true capabilities and limitations of the ADSs to the users to prevent misuse of the systems.

In order to establish the true capabilities and limitations of an Automated Driving Systems (ADSs), we need to first define their Operational Design Domain. An ODD refers to the operating environment (road type, weather conditions, traffic conditions) in which a vehicle can drive safely. For example, for Low-Speed Automated Driving (LSAD) systems such as pods and shuttles, the ODD may include urban areas with predefined routes that include pedestrians and cyclists. On the other hand, for a motorway chauffeur system, an ODD may include a four-lane divided motorway and dry conditions only. The types of scenarios a vehicle may encounter will be a function of its defined ODD, making this fundamental to any safety evaluation and scenario identification.

- A more formal definition of ODD as defined by SAE J3016 (2018) states that "Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics". -

In order to enable stakeholders to share, compare and re-use ODD definitions, there is a need for standards to provide guidance to the stakeholders on both the attributes to be used for ODD definition and a format for defining the ODD using those attributes. BSI PAS 1883 (UK) provides a taxonomy for ODD. Additionally, ISO 34503 uses the taxonomy to provide a high-level definition format for ODD. While these standardization activities address the important needs of the industry, a gap still exists in the industry for an ODD definition format for simulation.

ASAM OpenODD is a representation of the abstract ODD specification in a more well-defined syntax and semantics which enables machines to interpret and perform the required analysis. Additionally, the ASAM OpenODD specification shall be measurable and verifiable for the attributes it specifies.

2 Overview / Goals

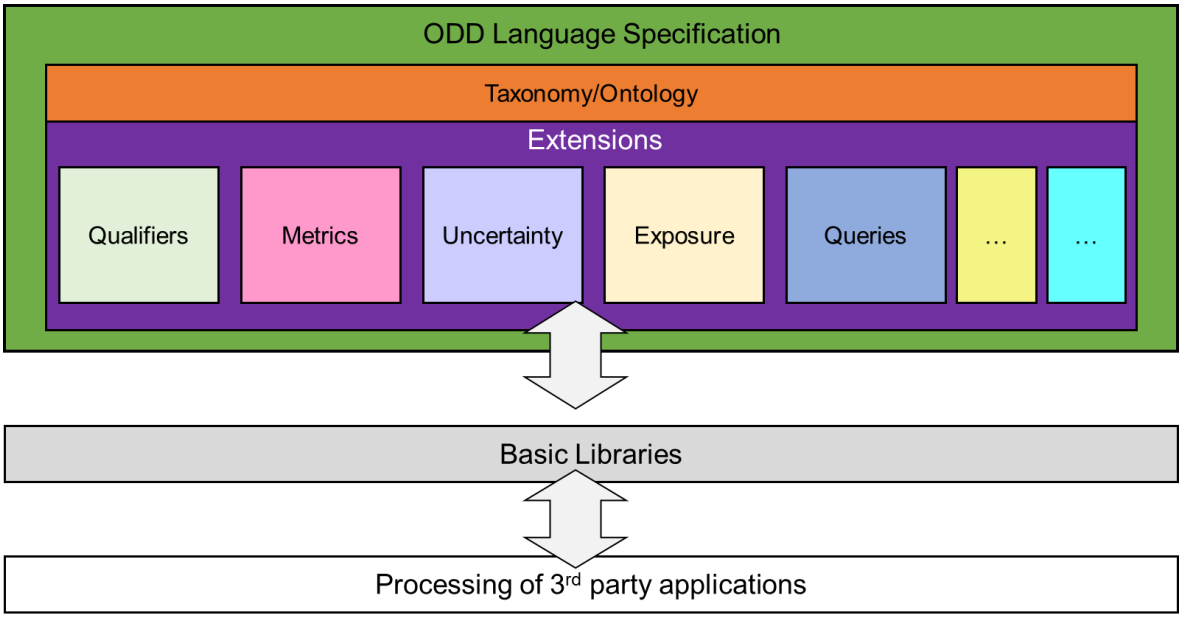
2.1 Motivation

An Operational Design Domain definition (ODD) shall be valid throughout the whole vehicle's lifetime. The definition of an ODD is part of the safety concept of a vehicle. Depending on the current development step different information needs to be derived from such an overall ODD.

ASAM OpenODD focuses on a machine-readable format. The ODD must be represented so it can easily be used within simulation and other machine processed environments. The content of ASAM OpenODD will be derived from any abstract ODD, providing the information in a usable manner. For the purpose of using this ODD description for simulations and post-processing the format must fulfill the following requirements:

- Searchable
- Exchangeable
- Extensible
- Machine readable
- Measurable and verifiable
- Readability (however, the use is for computer processing)

In the scenario based testing workflow ASAM OpenODD will play a very important role supporting the test description, defining the boundaries of what to test and achieving a good test coverage of the operational design domain and its borders.



The ODD language specification will include a Taxonomy/Ontology. The specification further includes concepts of qualifiers, metrics, uncertainty, exposure, queries and potential future extensions. However, the focus of the ASAM OpenODD project is on the language specification with the extensions. Taxonomy / Ontology will not be developed in the ASAM OpenODD project. Additionally, the choice of extensions to be handling in version 1 of ASAM OpenODD standard will be finalized in work package 3.

2.2 User stories

Development Engineer

- as a development engineer, I need to understand the relevant parameters and their ranges that can occur during automated driving within the ODD and the scenarios the automated driving system might be confronted with, and be able to trace my requirements to this kind of input data.
- as a development engineer, I need clear criteria how to detect at runtime that I am unintentionally leaving the ODD so that I can implement detection and reaction functions for this case.
- as a development engineer, I need input (e.g. List) on foreseeable situations in the ODD to test my sensor setup and software algorithms appropriately before vehicle simulation and road testing even starts.
- as a development engineer, I desire a machine readable ODD description that support the systematic segmentation of the ODD in sub-areas where different capabilities of the AD system are needed or different grades of AD functions are safe to operate.
- as a development engineer I want to check tests, scenarios, recordings, etc. against the ODD description.

- as a development engineer I want to use multiple inputs (e.g. from different third party vendors) to define the ODD.
- as a development engineer I want to provide an ODD description to other stakeholders such as Infrastructure operator (regulator, authorities).

Test Engineer

- as a test engineer, I would like to be able to declare an ODD to test against
- as a test engineer, I would like to know that a scenario that I have created is within the ODD
- as a test engineer, I would like to understand which element of my driving ontology/domain model is in or out of my ODD
- as a test engineer, I would like to know if the ODD that I have defined is well-formed and non-contradictory
- as a test engineer, I want to check the coverage of all my test cases against the given ODD.

Tool Developer

- as a tool developer, I would like to have a common format that I can share to define an ODD.
- as a tool developer, I would like the ODD format to be formal enough that I can execute and process the containing information.
- as a tool developer, I would like to be able to substitute multiple driving ontologies and still use the same ODD Format.
- as a tool developer, I would like to test my test bench capabilities against the ODD, with the goal if the capabilities suite the ODD needs.

Scenario editor

- as a scenario editor, I want to create the scenarios for an ADS which are within the defined ODD of the ADS.
- as a scenario editor, I want to create scenarios to test the ODD attribute boundaries to test the fall-back or Minimal Risk Manoeuvre capability of the ADS.
- as a scenario editor, I want to create scenarios at different levels of abstraction.

Data annotation Engineer

- as a label annotator I need to assemble a data library which is relevant to the ODD so the data (recordings, synthetic sensor data, etc) can be annotated accordingly
- as a label annotator I need to analyze data to identify ODD related situations and occurrences so that I can annotate them.
- as a label annotator I need to estimate the completeness of data to be annotated relative to the ODD so that users of the annotations can estimate achievable coverage.
- as a label annotator I need to estimate the coverage of annotations relative to the ODD so that users of the annotations can estimate coverage achieved.

Data Scientist
OpenODD Data Scientist Use-Cases, **extending** the ontology (i.e. need to support ontology by reference)

- as a data scientist, given a scenario library, I need to determine the parameter distribution, and coverage, so that I can assess the confidence in the results and deliver trust.
- as a data scientist, given a scenario library, I need to determine which aspects of the ODD are covered, so that I can assess which scenarios need to be added to support effective ODD testing.
- as a data scientist, given an ODD I need to determine whether a specific scenario is included in the ODD, so that I can determine whether it needs to be included in the assessment of results.
- as a data scientist, given an ODD, I need to determine the **exposure** to a specific scenario so that I can determine the weight/contribution of the scenario results in the overall assessment.
- as a data scientist, given two versions of the ODD I need to determine which scenarios were added or removed, so that I can determine how the analysis needs to be adapted.
- as a data scientist, given an ODD, I need to determine whether an **occurrence** is allowed by the ODD, so that I can determine whether the vehicle behaved as per the ODD specification.
- as a data scientist, given an ODD, I need to determine the exposure to a specific "occurrence" so that I can determine the weight/contribution of the behavior results in the overall assessment.

Safety Engineer

- as a safety engineer I want to (semi-automatically) derive a situation catalog for Hazard Identification from the formal ODD definition.
- as a safety engineer I want to demonstrate the metrics of coverage of my considered scenarios w.r.t. the total ODD and estimate the remaining risk of hazardous scenarios within the ODD that were not covered by my safety concept and my validation approach.
- as a safety engineer I want to identify scenarios of unintentionally leaving the ODD from a formal definition of the limits of the ODD, in order to handle these situations using appropriate detection and reaction possibilities.
- as a safety engineer I want to get an understandable representation which partitions or sub-areas of the ODD exist, which failures or degradations could lead to hazards in each of them and grades of AD functions or which behaviors ensure acceptably safe operation in each of them.

Infrastructure operator (regulator, authorities)

- as an infrastructure operator, I want to have easy access to ODD specifications in order to have a clear view on current and future demands on my road network

- as an infrastructure operator, I want to have the ability to warn vehicles in advance if they are about to chose a route through my road network which most likely is exceeding their ODD
- as an infrastructure operator, I want to give recommendations to automated vehicles which routes/roads/lanes to use depending on their ODD capabilities
- as an infrastructure operator, I want to have the ability to provide current ODD restrictions of parts of my network easily
- as an infrastructure operator, I want to be able to have a version controlled ODD Format
- as a regulator, I want to have a human-readable and exchangeable ODD format
- as a regulator, I want the ODD to be expandable when new regulations are introduced

While an Infrastructure operator may not be the direct user of the OpenODD format (due to the format's main use cases being for computer processing), it is important to incorporate requirements on attributes and language into the OpenODD format.

2.3 Requirements

Nr.	Cluster	Description
CL2	Machine readability and query	The language needs to be machine readable, can be parsed using any text parsers, should enable queries for attributes of the ODD. UTF8 Test using DSL,XML,YAML,others.
CL3	Composability	The language shall support combining/re-using multiple ODD definitions into a new, wider ODD, and it shall provide the ability to create a named subset of ODD attributes
CL4	Parameterization and Templating	In order to enable flexibility in definition, and flexibility in usage for testing and definitions, as well as the ability to supply templates for ODD definitions for various usages , the language should enable a parametrization mechanism, where names are replacing values, and are resolve toward actual usage time.
CL5	Conditional statement	Language specification needs to support conditional statement or reduced ODD. This can be tackled by imposing constraints on the full operational range of specific attributes.
CL6	Extensibility of Ontology	The language shall support the addition of new attributes and values that are not in the original ontology. It shall support the establishment of standard libraries, e.g. PAS 1883.

CL7	Binary boundary	The language shall enable mapping every situation into binary in/out of the ODD, convert 'uncertain' or 'fuzzy' situation into binary.
CL8	Class metric / datatype / units	class attributes must be separate from metrics, eg. the “rain” yes/no is separate from “droplet size”, the "speed limit”<100kph is separate from “road” class The language must support units mechanism. need to handle different datatype, enum, float, boolean, etc defining the unit is compulsory, unless it is a unit-less ratios.
CL9	Probability / Uncertainty / risk	The language shall support mechanism to represent uncertainty, probability and risk as optional element
CL10	Class Hierarchy / expressiveness / abstraction	The language format needs to be able to express the class hierarchy provided by the attribute group , it needs to support inheritance, be consistent with the class categories (keywords), data type and units. It shall support abstract ODD definitions by choosing different hierarchical levels.
CL11	Extensibility by function	The language shall be extensible by internal or external functions. Basic mathematical functions (like sqrt(), max(), min()) or functions with respect to the application domain (e.g. distance()) are required for conditions, limits and metrics. They can come from supplied library or external binding (e.g. C, Python)
CL12	Operators	The language shall use a set of mathematical and logical operators, such as +, -, and, or, >, <, etc
CL13	Permissive / Restrictive and other concepts	The language should take a mixed permissive and restrictive approach (and any other format concepts) to its statements of what is inside the ODD.
CL14	Integration in scenario based testing workflow	The language shall supply the ability to refer to ODD definition from scenario definition, interpretable to create a full test description , and can be used for different purposes (such as requirements, checking, success criteria or testing)
CL15	Compatible with OpenX and ISO Standards	The language shall be compatible with other OpenX and ISO standards

2.4 Relations to Other Standards, Projects or Organizations

Name of the Standard	Description
ISO 34503 (in development)	Provides a taxonomy for operational design domain for automated driving systems, along with a high level definition format for use by regulators and non-coders.
BSI PAS 1883	Provides a taxonomy that can be used to ODDs for automated driving systems
SAE AVSC00002202004 AVSC Best Practice for Describing an Operational Design Domain: Conceptual Framework and Lexicon	Aims to establish a best practice for Operational Design Domain description, establishing commonly defined terms and recommending a framework in which they can be applied.
SAE J3016 (2021) / ISO 22736	Defines the term ODD and its use for classifying levels of autonomy for automated driving systems
ISO 21448 SOTIF	Provides guidance on defining the safety of an ADS system or feature for its intended functionality.
ASAM OpenX standard suite	Provide naming convention of scenario and environment content,
ISO TS 5083 (in development)	Provides guidance on safety of an ADS system (design and verification stage).

3 Technical Content

3.1 Work Packages

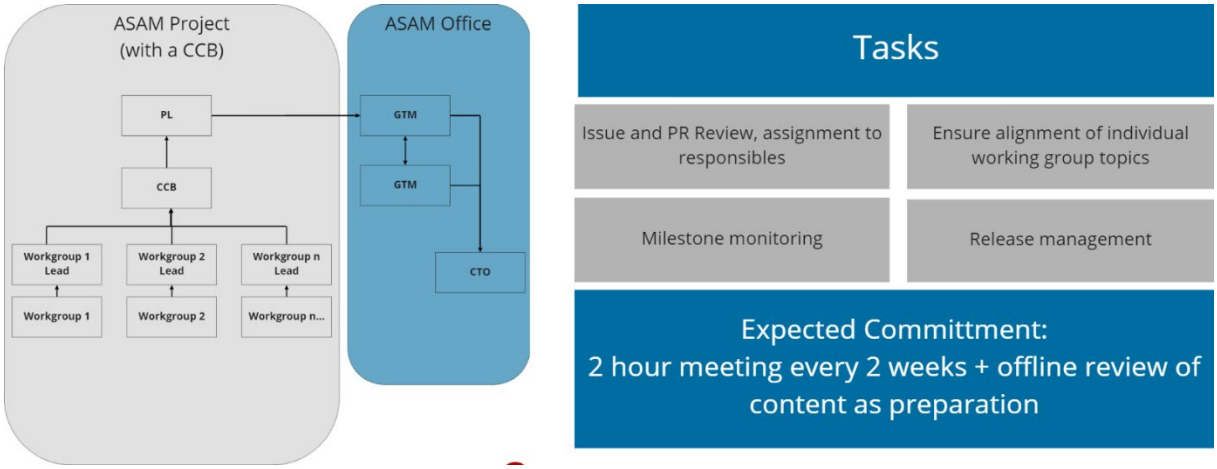
The following are the proposed work packages for the standardisation project:

3.1.1 WP1: Project coordination

This WP includes the overall WP for project management & coordination items. For the project a **Change Control Board (CCB)** will be setup. The CCB is a subgroup of experts in the project responsible for internal alignment of topics, release management & approval/review of pull requests.

The CCB will also be responsible for **Repository & issue maintenance**. They will interface with the ASAM office on project requirements for the CI/CD and other document related topics.

Furthermore, they will also perform the general issue maintenance – ensuring up to date issue overviews, working with the office & CCB on issue, milestone and progress reporting, including closing, assigning and pruning issues.



Duration: 17 months

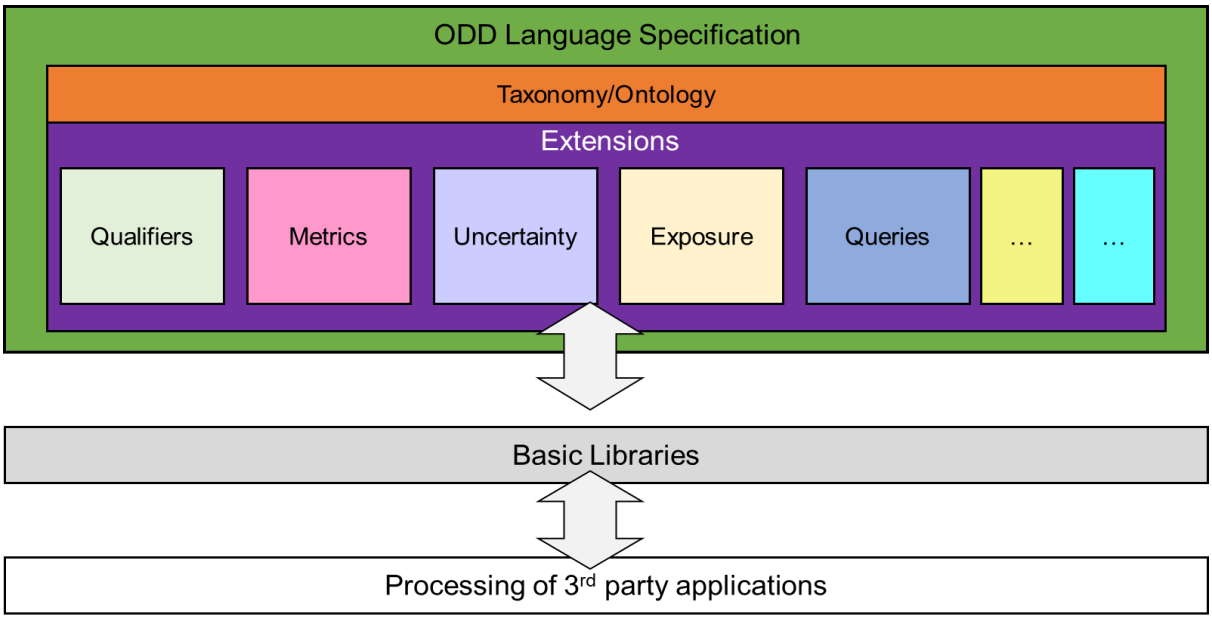
3.1.2 WP2: Ramp-up and knowledge sharing

The group is aware that there is a strong need to ensure a shared knowledge base of all participants for the content addressed in this project. This work package focuses on knowledge sharing and dissemination within the project group. **It is a mandatory WP for all participants.**

Duration: 2 months

3.1.3 WP3: Scope Refinement

This WP will review the current status of OpenODD scope and include scope finalization (relative to the ASAM OpenODD Concept Paper scope). The goal of this WP will include decision on including or excluding various aspects of the Figure below:



This will also include the scope definition for (considering the resource and time considerations):

- qualifier,
- metrics and
- ranking of the extensions,

Qualifiers represent qualitative probability in terms of upper bound on the occurrence rate. A metric is a function that transforms features or properties of an ODD - alone or in its relation to other ODDs, to scenarios etc. - into a number or an ordered set of qualifiers (e.g. good > medium > bad). It is not required that the function is mathematically defined and objective.

Deliverable: Finalised scope and the extensions to be handled by OpenODD v1.0 with a complete list of requirements. And a project plan with milestones every 2 month. Finalization of change control board.

On completion of this WP the project will share a detailed project plan and scope with the TSC. It is at this point that the TSC will be requested to evaluate resource commitments to WPs in relation to the project scope.

Duration: 3 months

3.1.4 WP4: Language

This WP will describe the semantic and syntactic description (including qualifiers) of the ODD description for format for simulation execution. It will include the capability to describe conditional ODD description using some or all ODD attributes.

Requirements based milestone will be put after the finalization of WP2.

Deliverable: Language definition (syntax and semantics)

Duration: 10 months + 3 month (writing) + 2 months (review)
[Note: for the case of selecting an existing language and building off it, the time line may be shortened significantly]

3.1.5 WP5: Extensions

This WP will describe the extensions for the ODD description format for computerized analysis of the usage of ODD definition. These may include uncertainty, probability, queries, metrics, dynamic assessment etc.

Rankings based milestone will be put after the finalization of WP2.

Deliverable: Format for extensions and format to link with language

Duration: 5 months + 1 month (writing) + 2 months (review)

3.1.6 WP6: Case studies

This WP will illustrate examples for each requirement of the OpenODD language.

Deliverable:

- Identifying case studies.
- Four ODD definition examples in the OpenODD v1.0 format

Duration: 1 (initial phase) + 2 months (end phase) + 1 month (writing) + 1 months (review)

3.2 Project Resources

Signing up to this project means a minimal commitment as follows:

1. **WP 2 Knowledge transfer and generation** – 2 month duration, current estimates are 1h meetings every week à 1 day
2. **WP 3 Scope Refinement** – 3 month duration, also 1h meeting every week + offline preparation à 3 days
3. **Later technical WPs** – 12 – 15 months duration – due to the need to perform additional scope refinement it is currently not feasible to estimate the necessary resources to finalize the standard. Part of the output of WP3 will be a refined list of participant commitments to the individual work items. For now companies are requested to provide a minimum commitment of 25 days.

A service provider is requested by the project for technical writing efforts. It is planned for such a partner to accompany the project from September (shortly after project start) onwards. This ensures a joint ramp-up with the project participants and a joint definition of project working mode (together with the responsible person from the ASAM office). The service provider will accompany the project until its completion to ensure quality of deliverables is maintained throughout all phases. Experience gained by ASAM in previous projects has shown a need for on average 20 hours per week effort by a technical writer. With a project duration of 78 weeks this entails 1.700 hours of effort total, or 215 days.

3.3 Company Commitments

Enrollments are open online. Company signups will remain open until shortly after the project kickoff.

3.4 Budget

Task description	Effort [person days]	Cost [EUR]
Technical writing	215	150.500

4 Review Steps

This project requests both an internal (project level) and a full public review (open to all, including non-ASAM members).

5 Deliverables

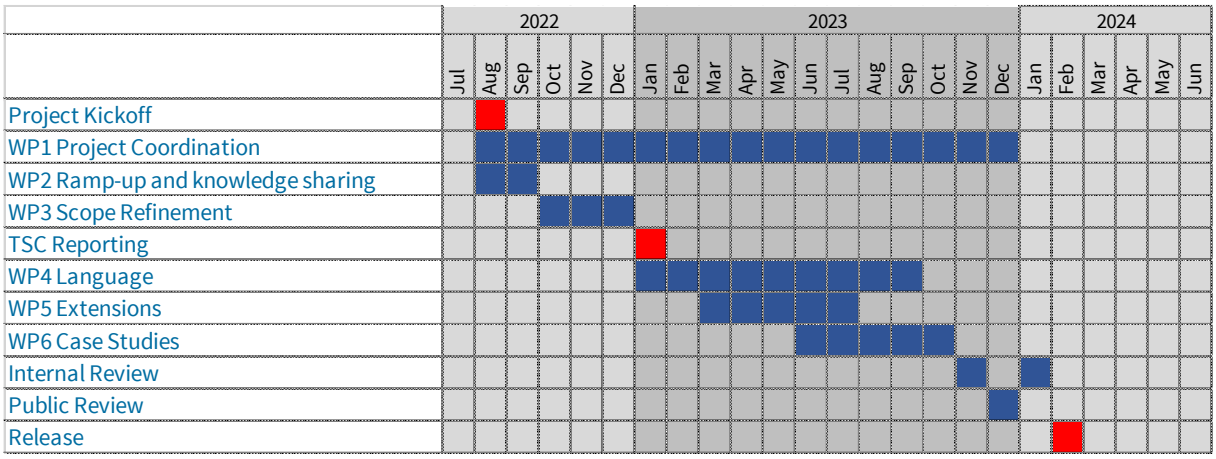
At the end of the project, the project group will hand over the following deliverables to ASAM:

Table 1 Deliverables

Item No.	Description
1	Standard specification document (normative)
2	User guide with examples that use the format (non-normative)

6 Project Plan

The work packages shall be carried out as per the following time schedule:




ASAM
Association for Standardization of
Automation and Measuring Systems