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	30.03.2024
ASAM Funds	
Proposer	Siddartha Khastgir (WMG), Andreas Richter (VW)

Table of Contents

1 Executive Summary 3

2 Overview / Goals 4

2.1 Motivation 4

2.2 User stories 5

2.3 Requirements 6

2.4 Relations to Other Standards, Projects, or Organizations 14

3 Technical Content 15

3.1 Work Packages 15

3.1.1 WP1: Project coordination 15

3.1.2 WP2: Ramp-up and knowledge sharing 15

3.1.3 WP3: Scope Refinement 16

3.1.4 WP5: Examples 16

3.1.5 WP6: Data Model 16

3.1.6 WP7: Database schema 17

3.1.7 WP8: Language 17

3.1.8 WP9: Case studies 17

3.1.9 Project member and public review 17

3.2 Project Resources 18

3.3 Company Commitments 18

3.4 Budget 19

4 Review Steps 20

5 Deliverables 21

6 Project Plan 22

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.

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 (e.g., road type, weather conditions, traffic participants) 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 a comprehensive machine- as well as human-readable ODD definition format for the development cycle of ADS from requirements engineering, development and testing (which heavily uses simulation for scenario-based testing), homologation and operation.

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

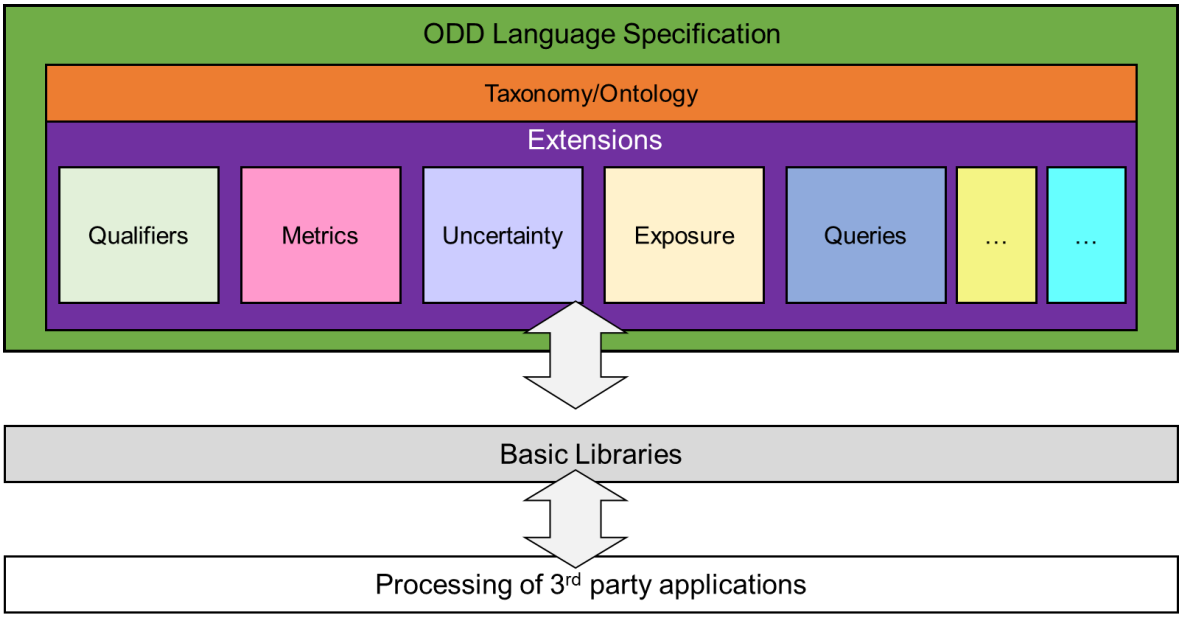
2.1 Motivation

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/query able
- Exchangeable
- Extensible
- Measurable and verifiable
- Machine readable
- 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 and providing value ranges of what to test and achieving a good test coverage of the Operational Design Domain and its borders.



The ODD language specification will be taxonomy so that any kind of taxonomy can be used OpenODD may provide guidelines for a preferred modeling of used taxonomies. The specification further includes concepts of qualifiers (such as metrics, uncertainty, exposure), queries and potential future extensions. However, the focus of the ASAM OpenODD project is on an abstract data model to define ODD as well Operational Domains (OD) and Current Operational Domains (COD) and implement this abstract data model into one database schema as well as one language specification. Taxonomy will not be developed in the ASAM OpenODD project.

2.2 User stories

OpenODD will cover the whole development of an ADS and therefore different requirements of stakeholders with various expectations for ODD specification. Stakeholders interested in ASAM OpenODD come from many different domains (OEMs, regulators, tool vendors, simulation, safety and many others) and have different needs. Thereof various user stories will be considered.

Development Engineer: get (e.g., a list of) relevant criteria and parameters with their value ranges to be able to detect unintentionally leaving of the ODD in a machine-readable description to check tests, scenarios, recordings, etc. against multiple ODD specification from different vendors and provide ODD specifications to other stakeholders such as Infrastructure operator (regulator, authorities).

Test Engineer: get a well-formed and non-contradicting ODD specification to determine, which elements are inside or outside the ODD, specify scenarios and declare tests against and check the coverage of test cases against the given ODD specification.

Tool Developer: get a common and formal format to substitute multiple driving ontologies to test the given capabilities of the ADS against the ODD specification.

Scenario editor: get an ODD specification to create scenarios inside and outside of ODD on different levels of abstraction to test e.g., fallback or Minimal Risk Maneuver capabilities of the given ADS.

Data annotation Engineer: get an ODD specification to build data libraries for annotate data (recordings, synthetic sensor data, etc.) according to the ODD specification and check completeness and coverage of the data relative to the ODD specification.

Data Scientist: want to determine ODD parameter distribution, coverage, exposure, and occurrence to check against scenario libraries and identify scenarios being inside or outside of ODD specification.

Safety Engineer: get a formal ODD specification with clear definitions of the limits to (semi-automatically) derive hazard identification to determine the risk of corresponding scenarios.

Infrastructure operator (regulator, authorities): get a human-readable and version controlled ODD specification that can be extended when new regulations are introduced to check demands towards infrastructure and provide information, recommendations, and warnings in advanced to the ADS.

2.3 Requirements

Based on the user stories, requirements towards the specification were derived in the first phase of the project. The requirement clusters of the previous ASAM OpenODD Concept project were incorporated into this requirements collection. The first phase of the project focused on the language only (following the ASAM OpenODD Concept Project) but during this development project it was decided to provide three deliverables (see chapter 5 Deliverables): an abstract data model (as core element of ASAM OpenODD), a database schema as well as a language (both as representation of the data model for data storage and maintenance as well as data provisioning) to satisfy all needs of the identified user stories (see previous chapter). The following matrix in **Fehler! Verweisquelle konnte nicht gefunden werden.** presents which requirements are to be considered in which one of deliverables.

☑: relevant for given work product

❒: not relevant

o: optional

?: still not decided

Table 1 Requirements Specification

ID	Category	Requirement text	Data model	DB schema	Language
oODD_REQ_1.1	Purpose	OpenODD shall provide capabilities to describe: • Operational Domain (OD) • Current Operational Domain (COD) • Target Operational Domain (TOD) • Operational Design Domain (ODD)	☑	☑	☑
oODD_REQ_1.2	Purpose	OpenODD shall implement "OD description statement" for description of Operational Domain and Current Operational Domain.	☑	☑	☑
oODD_REQ_1.3	Purpose	OpenODD shall implement "ODD description statement" for description of Target Operational Domain and Operational Design Domain.	☑	☑	☑
oODD_REQ_2.1	Taxonomy	Description of COD/OD/TOD/ODD shall be based on at least one taxonomy.	☑	☑	☑
oODD_REQ_2.2	Taxonomy	OpenODD shall provide capability to reference multiple taxonomies and their elements individually.	☑	☑	☑
oODD_REQ_2.3	Taxonomy	OpenODD shall provide semantics for referencing hierarchical taxonomy elements (on any level of hierarchy).	☑	☑	☑
oODD_REQ_2.4	Taxonomy	OpenODD shall support usage of taxonomy hierarchy in "OD description statements" and "ODD description statements".	☑	☑	☑
oODD_REQ_2.5	Taxonomy	At least one taxonomy element shall be used in each "OD description statement" or "ODD description statement".	☑	☑	☑

ID	Category	Requirement text	Data model	DB schema	Language
oODD_REQ_3.1.1	Machine readability	OpenODD shall have syntax defined.	☐	☐	☑
oODD_REQ_3.1.2	Machine readability	OpenODD shall have semantics defined.	☐	☐	☑
oODD_REQ_3.2	Machine readability	OpenODD syntax shall be of textual format.	☐	☐	☑
oODD_REQ_3.3	Machine readability	OpenODD semantics shall be unambiguous.	☐	☐	☑
oODD_REQ_3.4	Machine readability	OpenODD syntax shall include commenting feature allowing marking parts of COD/OD/TOD/ODD descriptions which are excluded from semantic interpretation.	☑	☑	☑
oODD_REQ_3.4.1	Machine readability	OpenODD shall provide syntax of commenting feature allowing relating it to: • the overall COD/OD/TOD/ODD definition • "OD description statements" • "ODD description statements"	☑	☑	☑
oODD_REQ_3.5	Machine readability	OpenODD shall provide formal grammar supporting querying of COD/OD/TOD/ODD description.	☑	☑	☑
oODD_REQ_4.1	Composa-bility	OpenODD shall provide capability to encapsulate full or partial definitions of COD/OD/TOD/ODD into named entities.	☑	☑	☑
oODD_REQ_4.2	Composa-bility	OpenODD shall provide capability to include encapsulated COD/OD/TOD/ODD definitions as named entities into given COD/OD/TOD/ODD definition.	☑	☑	☑

ID	Category	Requirement text	Data model	DB schema	Language
oODD_REQ_4.3	Composa-bility	OpenODD semantics of inclusion mechanism of encapsulated COD/OD/TOD/ODD definitions shall be unambiguous.	☐	☐	☑
oODD_REQ_5.1	Parametri-zation and templating	OpenODD shall provide capability to define named parameters which can be resolved into concrete value.	o	☑	☑
oODD_REQ_5.2	Parametriz-ation and templating	OpenODD shall provide capability to use defined parameters in: • another parameter definition • conditional statements • "OD description statements" • "ODD description statements"	o	☑	☑
oODD_REQ_6.1	Conditional statement	OpenODD shall provide capabilities to express conditions under which given "ODD description statement" is interpreted semantically.	?	☑	☑
oODD_REQ_7.1	Binary boundary	OpenODD shall ensure "ODD description statement" can be resolved into Boolean value (true or false).	?	?	☑
oODD_REQ_8.1	Datatypes and units	OpenODD shall provide set of data types applicable in COD/OD/TOD/ODD definitions.	☑	☑	☑
oODD_REQ_8.1.1	Datatypes and units	OpenODD shall provide methodology for using external data types in referenced taxonomy elements within COD/OD/TOD/ODD definitions.	☑	☑	☑
oODD_REQ_8.1.2	Datatypes and units	OpenODD shall enforce assignment of data type for every referenced taxonomy elements.	☑	☑	☑

ID	Category	Requirement text	Data model	DB schema	Language
oODD_REQ_8.1.3	Datatypes and units	OpenODD shall enforce assignment of data type for every defined parameter.	☑	☑	☑
oODD_REQ_8.3	Datatypes and units	OpenODD shall conform to ASAM Unit Handling standard v. 1.0.0.	☑	?	☑
oODD_REQ_8.2	Datatypes and units	OpenODD shall enforce unit and unit type (as per ASAM Unit Handling) being assigned to every taxonomy element or defined parameter.	☑	☑	☑
oODD_REQ_9.0	Expressions and operators	OpenODD syntax shall conform to ASAM AE Expressions standard v. 1.0.1.	?	?	☑
oODD_REQ_9.1	Expressions and operators	OpenODD shall define mathematical operators, their operands, syntax and semantics.	☐	☑	☑
oODD_REQ_9.2	Expressions and operators	OpenODD shall implement at least following mathematical operators expressing semantics of: • addition • subtraction • multiplication • division	☐	☑	☑
oODD_REQ_9.3	Expressions and operators	OpenODD shall define relation operators, their operands, syntax and semantics.	☐	☑	☑
oODD_REQ_9.4	Expressions and operators	OpenODD shall implement at least following relation operators expressing semantics of: • equals • not equal • greater than	☐	☑	☑

ID	Category	Requirement text	Data model	DB schema	Language
		• greater than or equal • less than • less than or equal			
oODD_REQ_9.5	Expressions and operators	OpenODD shall define logical operators, their operands, syntax and semantics.	☐	☑	☑
oODD_REQ_9.6	Expressions and operators	OpenODD shall implement at least following logical operators expressing semantics of: • negation • logical conjunction • logical disjunction	☐	☑	☑
oODD_REQ_9.7	Expressions and operators	OpenODD syntax and semantics shall provide means of grouping logical expressions.	☐	☐	☑
oODD_REQ_9.8	Expressions and operators	Grouped logical expression shall be possible to use in another logical expression as one of operands.	☐	☐	☑
oODD_REQ_9.9	Expressions and operators	OpenODD shall define ODD boundary operators, their operands, syntax and semantics.	☐	☐	☑
oODD_REQ_9.10	Expressions and operators	OpenODD shall implement at least following ODD boundary operators expressing semantics of: • inclusion • exclusion	?	☑	☑
oODD_REQ_10.1	Global mode	OpenODD shall provide means to handle taxonomy elements not referenced by any "ODD description statements" in	☑	☑	☑

ID	Category	Requirement text	Data model	DB schema	Language
		ODD description with at least following semantics: inclusion, exclusion, indifference.			
oODD_REQ_11.1	Traceability from external systems	OpenODD shall provide syntax and semantics for unique identification of referenced taxonomies and their elements.	☑	☑	☑
oODD_REQ_11.2	Traceability from external systems	OpenODD shall provide syntax and semantics for unique identification of "OD description statements" used in COD or OD definition.	☑	☑	☑
oODD_REQ_11.3	Traceability from external systems	OpenODD shall provide syntax and semantics for unique identification of "ODD description statements" used in TOD or ODD definition.	☑	☑	☑
oODD_REQ_12.2	Custom user data and labeling	OpenODD syntax shall provide capability to mark parts of COD/OD/TOD/ODD description dedicated for custom user data and excluded from semantic interpretation.	☑	☑	☑
oODD_REQ_12.1	Custom user data and labeling	OpenODD syntax shall provide capability to relate custom user data to: the overall COD/OD/TOD/ODD definition • "OD description statements" • "ODD description statements"	☑	☑	☑
oODD_REQ_13.1	Timestamp and location	OpenODD shall provide a mechanism to state time and location for: • the overall COD/OD definition • OD description statements	☑	☑	☑

ID	Category	Requirement text	Data model	DB schema	Language
oODD_REQ_13.2	Timestamp and location	OpenODD shall implement at least following ways to express time: • point in time (timestamp) • timespan	☑	☑	☑
oODD_REQ_13.3	Timestamp and location	OpenODD shall implement at least following ways to express location: • single point • area	☑	☑	☑

2.4 Relations to Other Standards, Projects, or Organizations

Table 2 relevant standards

Name of the Standard	Description
ISO 34503 (in development)	Provides a taxonomy that can be used in ODDs 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 in ODDs for automated driving systems
SAE J3259 (in development)	Provides a taxonomy that can be used in 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 ODD 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, type and schema definitions
ISO TS 5083 (in development)	Provides guidance on safety of an ADS system (design and verification stage).

3 Technical Content

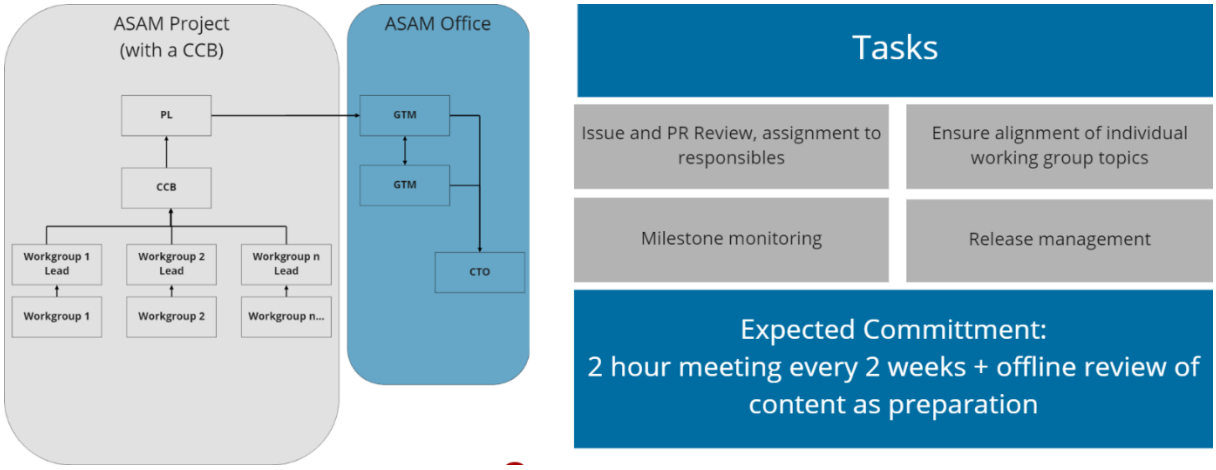
3.1 Work Packages

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

3.1.1 WP1: Project coordination

This WP includes the overall WP for project management & coordination items. As of July 2023, active development has not begun, the project has defined subgroups and alignment takes place between PLs and subgroup leads. Once active development on the deliverables begins, a Change Control Board (CCB) shall be set up. 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 wasn't established until the subgroup work started. The tasks of the CCB are currently covered by sync meetings of the subgroup leads as well as syncs together with the project leads.

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 to gain a common understanding of the topic and the goals of ASAM OpenODD. In other words: everybody is

talking about the same. **It is a mandatory WP for all participants** and open for every interested company.

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). Furthermore, it will be checked if additional requirements must be considered, which were not covered in the previous concept project. The goal of this WP will include decision on including or excluding various aspects from the concept paper as descripted in chapter 2.1 Motivation (for more details see ASAM OpenODD Concept Paper).

Deliverable: Finalized scope and the extensions to be handled by OpenODD v1.0 with a complete list of requirements. Additionally, the project plan with milestones will be updated.

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 WP5: Examples

After gaining a common understanding (WP2) and agreeing on the scope (WP3) the project members will create examples for describing relevant assets of OpenODD (such as Current Operational Domain (COD), Operational Domain (OD), Operational Design Domain (ODD) as well as representation of the used taxonomy) to derive additional needs for the specification.

Duration: 3 months

3.1.5 WP6: Data Model

Based on the initial requirements from WP3 and additional needs demonstrated in WP5 a first draft of a data model for OpenODD will be defined. This data model will cover the relevant assets COD, OD, ODD and taxonomy. This data model is input for the following WP and will incorporate their findings to update the OpenODD data model.

Deliverable: abstract data model

Duration: 6 months

3.1.6 WP7: Database schema

The data model from WP5 will be implemented as database schema. This should enable OpenODD users to manage OpenODD compliant assets in databases. This WP will describe the semantic and syntactic description (including qualifiers) of the ODD description for database applications such as management of ODDs or provisioning for validation and verification use cases.

Necessary adaptions for the database and general findings will be feed back to the data model WP.

Deliverable: Database schema

Duration: 6 months

3.1.7 WP8: Language

The data model from WP5 will be implemented as a language. This should enable OpenODD users to exchange OpenODD compliant assets as files. This WP will describe the semantic and syntactic description (including qualifiers) of the ODD description for format for data exchange e.g., necessary for simulation execution or provision to third party stakeholders such as public authorities.

Necessary adaptions for the language and general findings will be feed back to the data model WP.

Deliverable: Language definition (syntax and semantics)

Duration: 6 months

3.1.8 WP9: Case studies

This WP will illustrate examples for each requirement of the OpenODD specification covering database schema as well as language.

Deliverable:

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

Duration: 4 months

3.1.9 Project member and public review

The project will conclude with a project member review of the specification document as well as of all deliverables. After approval by the project members a public review will take place to collect last findings and suggestions.

Duration: 2 months project member review + 2 months public review

3.2 Project Resources

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

1. **WP2 Ramp-up and knowledge sharing** – 2 months duration, current estimates are 2h meetings every week → 1 day
2. **WP3 Scope Refinement** – 3 months duration, also 2h meeting every week + offline preparation → 3 days
3. ***WP5 Examples*** – 3 moth duration, 2h meeting every second week + offline preparation → 3 days
4. **Later technical WPs (6, 7, 8 as well as WP9 Case studies)** – 4 – 6 months duration – 2h bi-weekly meeting + offline work each. Due to the interleaving of the work packages as well as different complexity of tasks the necessary effort will vary. Additionally, not every project member may have to contribute to all of these work packages. 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 and management of the abstract data model. It is planned for such a partner to accompany the project from Summer (shortly after first draft of the data model is ready) 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 15 weeks this entails 609 hours of effort total, or 77 days.

3.3 Company Commitments

Commitments by project members after the decision to expand the scope (December 2022). Status December 2022: Average commitment of 26-man days per person.

Work package Commitments:

- **WP 5: Examples & WP6 Data Model**
  Required: 112,5 person days
  Committed: 130 person days
- **WP 7: Database Schema**
  Required: 150 person days
  Committed: 242 person days

- **WP8: Language**
  Required: 195 person days
  Committed: 258 person days
- **WP9 Case Studies & project member & public review**
  Required: 225 person days
  Committed: 225 person days

Overall required: 682.5
Overall committed: 855

3.4 Budget

Task description	Effort [person days]	Cost [EUR]

4 Review Steps

This project requests both a project member 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:

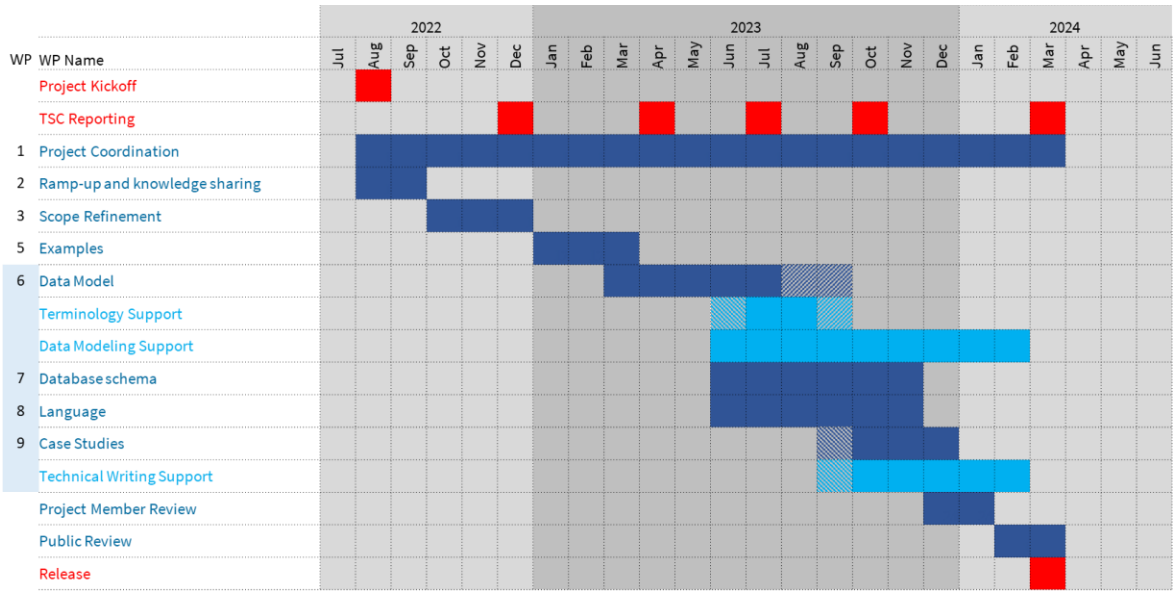
Table 4 Deliverables

Item No.	Description
1	Standard specification document (normative) including: • Abstract data model • Database schema • Language definition
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:

Table 5 Time schedule




ASAM
Association for Standardization of
Automation and Measuring Systems