Rail to Digital automated up to autonomous train operation

D35.3 – Analysis Tools specification

Due date of deliverable: 31/05/2026

Actual submission date: 06/11/2025

Leader/Responsible of this Deliverable: Antoine Legros - SNCF

Reviewed: Y

Document status		
Revision	Date	Description
01	06/06/2025	First version updated after first review by task participants
02	03/10/2025	Update after second review by task participants
03	24/10/2025	Update after WP review
04	07/07/2026	Update after JU review
05	08/10/2026	Update from Sensitive to Public

Project funded from the European Union's Horizon Europe research and innovation programme		
Dissemination Level		
PU	**Public**	**X**
SEN	Sensitiv – limited under the conditions of the Grant Agreement	

Start date: 01/12/2023 Duration: 30 months

ACKNOWLEDGEMENTS



This project has received funding from the Europe's Rail Joint Undertaking (ERJU) under the Grant Agreement no. 101102001. The JU receives support from the European Union's Horizon Europe research and innovation programme and the Europe's Rail JU members other than the Union.

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Antoine Legros	SNCF	Structure of the document. Sections 1, 2, 5, 7 and 8. Review.
Ernesto Virseda Alvaro Miguel Fernández Elorriaga Rodrigo Caceres Harris	CEDEX	Section 3. Review.
Diana Serra Rosario Esposito Salsano	FSI	Section 4.
Alessandro Caiazza	HITACHI	Section 6.1.
Paolo Stoico	MERMEC	Section 6.2 and Appendix 1. Review.
Angel Fernandez Gago	ADIF	Review.
Thomas Waschulzik	SMO	Review.
Jonathan Schöner	DLR	Review.

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author’s view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

1 EXECUTIVE SUMMARY

The task 34.3 delivered the specifications of the tests facilities [2]. This document mainly focused on the interfaces between the system under test and the test bench but it also proposed first insights into the analysis tools that should be used during tests campaigns.

Analysis tools used for test evaluation are critical for certification tests campaigns and therefore must fulfill some functional and non-functional requirements.

The main task of the analysis tools is to automatically evaluate the result of the tests. Therefore, the main part of this document is dedicated to the concept of test evaluation.

First, this deliverable defines the generic concept of test evaluation that the analysis tools must be able to address for both Verification/Validation and performance tests. Each step of the test must be evaluated according to the expectations of the test. Based on the steps evaluations, the result of the overall test can only be Passed if none of the steps of the test are Non-Passed or associated to a Test Error. As soon as one of the steps is Non-Passed, the overall test is considered as “Non-Passed” as well. Otherwise, the result of the overall test mentions the fact that a Test Error, which affects the initial target of test, is present.

This deliverable also details the different types of checks that can be specified by a test step:

- Event-based checks are checks following an event which is supposed to happen only once at a specific distance and time during the test. Due to the uncontrollable events that can occur during the test run and because some reactions of the system under test can only be estimated in terms of time and distance by the test, validation ranges (in time or distance) can be associated to this kind of check. The step is considered as “Passed” if the event occurs in the time or distance range around the test step.
- Continuous checks are performed repeatedly throughout the entire test, without limitations on time or distance, with or without additional conditions based on other outputs.
- Checks on time/distance windows are performed repeatedly on a limited time or distance window during the test.

Then this deliverable applies these generic definitions to specific subsystems and interfaces to give more concrete and functional examples of what is expected from the analysis tools regarding the test evaluation. As in several others tasks of WPs 34 and 35, the focus is made on ASTP but other functions like ATO, the Digital Register and the Onboard Communication Network are also considered to enlarge the scope of potential checks.

The main output of the test evaluation is the test report. This document must contain several information regarding the test environment, the system under test and of course the results of the test. The expectations for the test report may differ between validation test campaigns and certification test campaigns: depending on the context, some information or restrictions for the test campaign can be required by NoBos. The annex of this deliverable provides a template for the test report.

Besides the automatic test evaluation itself, the analysis tools can also be useful for other aspects, like allowing the testing team to manually check the test results or investigating potential issues in the system under test. Prior to the automatic test evaluation, the analysis tools must also perform some actions to prepare it, like for instance logging all the inputs and outputs of the system under

test. Some functionalities of the analysis tools are performed at the same time as the test execution, whereas others are performed in post treatment. All the functionalities that are expected from the analysis tools are specified in the form of functional and quality requirements.

ABBREVIATIONS AND ACRONYMS

AD	Automatic Driving
ASTP	Advanced Safe Train Positioning
ATO	Automatic Train Operation
ATO-AT	Automatic Train Operation Transactor
BG	Balise Group
BTM	Balise Transmission Module
CENELEC	European Committee for Electrotechnical Standardization
CCS	Control Command and Signalling
CMD	Cold Movement Detection
DMI	Driver Machine Interface
DR	Digital Register
ERJU	Europe's Rail Joint Undertaking
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
GoA	Grade of Automation
GUI	Graphical User Interface
IM	Infrastructure Manager
JRU	Juridical Recording Unit
KPI	Key Performance Indicator
LTM	Loop Transmission Module
MBS	Moving Block System
NoBo	Notified Body
OB	On Board
OCORA	Open CCS On-board Reference Architecture
ODO	Odometry
RAMSS	Reliability, Availability, Maintainability, Safety and Security
RCA	Reference CCS Architecture
RIU	Radio Infill Unit
RTM	Radio Transmission Module
RU	Railway Undertaking

R2DATO	Rail to Digital automated up to autonomous train operation
STM	Specific Transmission Module
S2R	Shift2Rail
TCMS	Train Control Management System
TDA	Train Data Acquisition through external sources
THR	Tolerable Hazard Rate
TIU	Train Interface Unit
TMS	Traffic Management System
TS	Trackside
UTC	Universal Time Coordinated
WP	Work Package

TABLE OF CONTENTS

Acknowledgements 3
Report Contributors 3
1 Executive Summary 5
Abbreviations and Acronyms 7
Table of Contents 9
List of Figures 11
List of Tables 11
2 Introduction 12
3 Generic concept of Test Evaluation 14
3.1 General information 14
3.1.1 Verification/Validation tests 14
3.1.2 Performance tests 14
3.2 Steps classification 14
3.2.1 According to the Interface Nature 15
3.2.2 According to the Interface Category 15
3.3 Test evaluation categories 15
3.3.1 Result Categories for Test Steps 15
3.3.2 Result Categories for overall Test 18
3.4 Evaluation rules 19
3.4.1 Examples of Specific Evaluation Rules: 20
3.4.2 Evaluation of Dependent and Independent Steps in Case of Special Step Results 20
4 Generic types of checks 21
4.1 Event-based checks 21
4.1.1 Concept of validation ranges 21
4.1.2 Time ranges 21
4.1.3 Distance ranges 21
4.2 Continuous checks 22
4.3 Checks on time/distance windows 22
5 Examples of application and adaptation of evaluation rules for some interfaces 23
5.1 Advanced Safe Train Positioning 23
5.1.1 Verification and Validation tests 23
5.1.2 Performance tests 24
5.2 Automatic Train Operation up to GoA2 27
5.2.1 Verification and Validation tests 27

5.2.2 Performance tests 29
5.3 Digital Register 30
5.3.1 DR-TS 31
5.3.2 DR-OB 32
5.4 Onboard Communication Network 33
6 Test report 35
6.1 Content 35
6.2 Template 36
7 Requirements for analysis tools 38
7.1 Tools running during the test 38
7.1.1 Functional requirements 38
7.1.2 Quality requirements 39
7.2 Tools running in post treatment 41
7.2.1 Functional requirements 41
7.2.2 Quality requirements 44
8 Conclusions 47
References 48

LIST OF FIGURES

Figure 1: Evaluation of the overall test 19

LIST OF TABLES

Table 1: Result Categories examples 18
Table 2: Dependent and independent steps 20

2 INTRODUCTION

The present document constitutes the Deliverable D35.3 “Analysis Tools specification” in the framework of the WP35 of FP2 R2DATO.

Activities within WP35 are the continuation of what has been started with WP34. Results and outcomes of WP34 are considered as inputs for the implementation of the new activities.

WP34 focused on defining preliminary strategy, requirements and a first set of specifications and architecture for the tests facilities.

The main objectives of the WP35 are to:

- Update the testing strategy according to the progress and outcomes of the System Pillar regarding architecture (Task 35.1).
- Define the certification methodology for the CCS+ domain, including on-board and trackside (Task 35.2).
- Define analysis tools to apply common methods for interpreting test results (Task 35.3).
- Implement and realise a test platform supporting virtual certification to support the strategy and the processes defined as part of the TE (Task 35.4).
- Define the processes for validation and virtual certification of digital twins of subsystems and functionalities apart from CCS+ (Task 35.5).

This deliverable focuses on the analysis tools that will be required within the tests facilities to implement and support the testing and certification methodologies developed by the other tasks of this WP35.

The content of this deliverable is based on the outcomes of WP34, in particular the specifications of the tests facilities delivered by Task 34.3 [2], but also on other previous and current works regarding certification like the SUBSET-076 [7] development.

Chapter 3 describes the generic concept of test evaluation that analysis tools will have to handle. It classifies the different kind of steps that may be encountered within test sequences and indicates the different results categories which are available to evaluate them.

Chapter 4 specifies the generic types of checks that can be required by test sequences. It introduces the notions of event-based checks and their possible validation ranges, as well as continuous checks or checks performed on time/distance windows.

Chapter 5 gives examples of application and adaptation of evaluation rules for some interfaces, focusing on the ASTP (as a continuity of the work done in WP34), ATO, Digital Register and Onboard Communication Network. Distinction is made between verification and validation tests and performance tests. Several examples of checks are described for each type of test and these checks are classified with the categories introduced by Chapter 4. This chapter relies on the outcomes of WPs 21, 22, 23 and 27 to apply the generic principles developed in the previous chapters for the ASTP, the Onboard Communication Network and the Digital Register. For the ATO (up to GoA2), the main inputs are the CCS TSI 2023 [11] and in particular the SUBSET-125 [10].

Chapter 6 specifies the expected content for a certification test report. A template for the test report is provided as an Annex to this deliverable.

Chapter 7 summarizes the content of the previous chapters from a formal point of view. It specifies all the requirements that the analysis tools, needed within the certification test platform, must fulfill. Requirements are classified in two categories: functional and quality requirements. Tools running during the test or in post treatment are not submitted to the same specifications so they are treated in different sub-sections. For each requirement, it is also indicated whether it is mandatory (“must have”) or optional (“nice to have”).

Finally, Chapter 8 provides the final conclusions and an overview of the main concepts introduced by this document.

3 GENERIC CONCEPT OF TEST EVALUATION

3.1 GENERAL INFORMATION

Current limitation of this document's version: *Assumptions regarding the test structure, which may differ depending on the system under test.*

The test evaluation shall be understood as the process where the test results for each test are checked against the reference test, or against the validation ranges defined for each type of result. The validation ranges are defined in 4.1.1.

This test evaluation depends on the type of tests that are envisaged for each system under test. The following sub-sections mention the different types of tests and their generic evaluation concept.

A preliminary, and sometimes comprehensive (depending on the considered Building Block), definition of the verification, validation and performance tests for some of the on-board and the trackside Building Blocks can be found, respectively, in the chapters 6 and 9 of D34.1 - General testing Strategy for a common virtual certification process [16] and updated in chapters 6 and 9 of D35.1 - General testing Strategy for a common virtual certification process [17].

3.1.1 Verification/Validation tests

This type of tests can be evaluated comparing the expected result defined in each step of the test with the step result obtained in the test. For example, for the ATP-OB, the SUBSET-076 [7] of the CCS TSI evaluates the functional behaviour of each step i.e. the expected result defined in each step of the tests sequence is compared with the step result of the test. The validation may consist of the verification of a specific event or that this event occurs within a validation range.

3.1.2 Performance tests

This type of tests involves, firstly, the definition of the performance features or characteristics (like the integrity rate, the accuracy, the availability, etc.), and secondly, the performance metrics (or Key Performance Indicators) defined for those features. So, statistical and graphical analyses will be carried out to evaluate these tests.

The main types of checks for these performance tests will be the continuous checks (after the execution of the test) as well as the checks on time/distance windows (see their definitions in 4.2 and 4.3, below). Validation ranges can also be used in performance tests.

3.2 STEPS CLASSIFICATION

Normally, the tests shall be defined by a list of steps completely defined with the corresponding description, additional information to take into account and the interface nature and category.

According to the interface nature and category the following sub-sections define the steps classification.

3.2.1 According to the Interface Nature

Test steps are categorized by event type and then by whether the event is an input or an output of the system under test.

3.2.1.1 Input Step

Input steps are the test events that reach and/or stimulate the system under test, so in fact they can be considered as outputs from the laboratory to the system under test, so these inputs steps shall be managed by the laboratory and/or the tester in charge of the test.

The evaluation of these steps consists of checking if the events have produced the corresponding input in the system under test defined in the test.

3.2.1.2 Output Step

Output steps are the test events to define the reactions of the system under test to the programmed inputs received.

The evaluation of these steps consists, firstly, of checking if the event has been produced by the system under test. And in second place, if the event result is within the validation range defined.

3.2.2 According to the Interface Category

Depending on the system under test, the test steps are classified according to the interfaces to be evaluated and/or the interfaces involved in the test. So, for each system under test, its own interfaces must be defined independently. For example, in SUBSET-076 [7] the interface classification contains the following interfaces: BTM, DMI, RTM, LTM, STM, TIU, ODO, TDA, ATO and CMD.

In this category classification, the types of results that will be evaluated in the test steps can also be defined.

3.3 TEST EVALUATION CATEGORIES

3.3.1 Result Categories for Test Steps

Once the Validation or Performance Test is executed, in the test evaluation, each step of the test can be classified according to the following step result categories:

3.3.1.1 Passed

This category applies to steps that have the result according to the test description. If it is an “Input” step it means that the event (from laboratory and/or tester to the system under test) is according to the test step description. If it is an “Output” step it means that the system under test behaviour is according to the test step description.

Also, this category applies to steps when the event and/or the behaviour occurs within the validation ranges defined for each step (see the sub-section 4.1.1 for the definition of the validation ranges).

This category can apply to steps of both validation and performance tests.

3.3.1.2 Passed with comments

This category applies to steps that have the result according to the test description, but the behaviour has some minor differences to the expected one (for example optional variables, different range of values...).

When a step is evaluated with this step result category, the comment describing why the step has been evaluated in this way, shall be included in the test evaluation.

This category can apply to steps of validation tests.

3.3.1.3 Non-Passed

This category applies to any step that has the result different to the test description. This category only applies to “Output” steps, and it means that the system under test behaviour is not in line with the test step description.

Also, this category applies when the behaviour occurs outside the validation ranges defined for each step.

This category can apply to steps of both validation and performance tests.

3.3.1.4 Linked to Previous Error

This category applies to any step that cannot be evaluated because a previous error (Non-Passed step) has occurred in the test and affects the following steps.

With this classification it is possible to distinguish between the source error and the inherited errors of previous steps.

This category can apply to steps of both validation and performance tests.

3.3.1.5 Optional Functionality

This category applies to any step of optional functionalities included in the tests but not implemented in the equipment under test. (For example, in SUBSET-076 [7]: STM, RIU, Euroloop and Train Data acquired from ERTMS/ETCS external sources functions are considered as “Optional Functionality”).

This category can apply to steps of validation tests.

3.3.1.6 Not applicable

This category applies to any step that has no associated interface in the test sequence or internal interface steps not relevant to verify the system under test conformity.

This category can apply to steps of both validation and performance tests.

3.3.1.7 Test error

This category applies to steps affected by detected errors in the test.

This category can apply to steps of both validation and performance tests.

3.3.1.8 Observation

This category applies to results of test or behaviours of the system under test that are not considered into the test description but are relevant for the test evaluation, although they fulfil all requirements of the specification of the system under test.

When a result is evaluated with this result category, the comment describing the behaviour of the system under test shall be included in the test evaluation.

This category can apply to steps of both validation and performance tests.

3.3.1.9 Deviation

This category applies to results of test or behaviours of the system under test that are not considered into the test description but are relevant for the test evaluation, because they don't fulfil some requirements of the specification of the system under test.

When a result is evaluated with this result category, the comment describing the behaviour of the system under test shall be included in the test evaluation.

This category can apply to steps of both validation and performance tests.

3.3.1.10 Result categories examples Table

The next table are just simple examples of each category. They are steps that could be in a real test, but here it must be considered just as an explicative example.

Step description	Result	Comments
(REQ: FP2-ASTP-SRS-038): ASTP computed estimated position accuracy for at least 95% of the cases, the absolute error of the ASTP reference estimated position provided by ASTP shall not exceed: +- 5.0 m in areas with negligible constraints.	Passed	It has been proved that this requirement is fulfilled: the threshold (+- 5.0m) has never been exceeded during the test.
From the train power on, the ASTP shall initialise itself and provide the outputs with no human supervision.	Passed with comments	The ASTP output from the train power on requires a driver validation. This behaviour is due to a new CR (change request), so the step is classified as Passed with comments
The computed safe ½ confidence interval toward estimated speed shall not exceed 2 km/h (MASO/MASU), for speeds lower than 30 km/h, and increasing linearly up to 12 km/h for speeds between 30 km/h and 500 km/h.	Non-Passed	When the speed= 25 km/h the confidence interval toward estimated speed is = 4 km/h

After Emergency brake has been commanded, the OB equipment releases the Emergency brake when the train reaches standstill.	Linked to previous error	Emergency brake is not commanded as expected in the test, therefore the step that checks later that the EB is released cannot be Passed and is Linked to Previous Error.
The OBU tries to connect with Radio Infill Unit.	Optional Functionality	RIU is not implemented as it is an optional function.
ERTMS/ETCS on-board unit is powered up.	Not applicable	NO interface exists in order to test that the OBU is powered up, it is a natural assumption.
The BG error is recorded.	Test error	No error has been introduced in the BG message in the test so no error is detected by the system under test.
FP2-ASTP-SRS-038: For at least 95% of the cases, the absolute error of the ASTP reference estimated position provided by ASTP (difference between the real position of the train simulated by the test bench and the estimated position provided by ASTP) shall not exceed +/-5m.	Observation	Although the absolute error does not exceed +/-5m, in 97% of the cases as expected, but in 2% of the cases the error exceeds 100m.
When the train reaches standstill, the Service brake is released.	Deviation	The Service Brake is released as expected but it is also noticed that all the track description that are in advance of the train position at standstill are deleted, which is not consistent with the specifications.

Table 1: Result Categories examples

3.3.2 Result Categories for overall Test

Once the Validation or Performance Test is executed, the result is computed based on the steps evaluations and can be classified according to the following result categories for the overall test:

3.3.2.1 Passed

This category applies if **none** of the steps have the result Non-Passed or Test Error.

3.3.2.2 Non-Passed

This category applies if **one or more** steps have the result Non-Passed.

3.3.2.3 Test error

This category applies if **one or more** steps have the result Test error and **none** of the steps have the result Non-Passed.

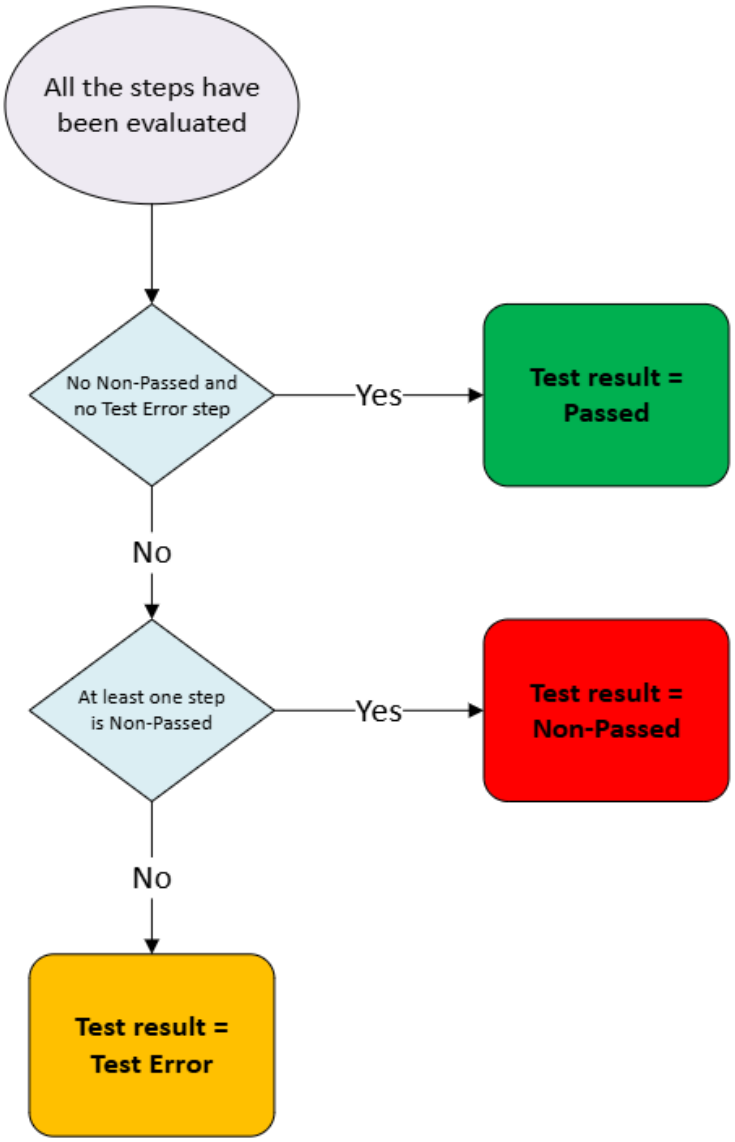


Figure 1: Evaluation of the overall test

3.4 EVALUATION RULES

The evaluation rules for the tests depend on the system under test, the involved interfaces, the type of test and the type or result to be verified.

As generic evaluation rule, the results of each step/test shall be evaluated according to the concrete specifications of its interface (if they exist) and/or the concrete requirements defined for the test.

3.4.1 Examples of Specific Evaluation Rules:

3.4.1.1 SUBSET-076 - Validation Test:

In SUBSET-076 [7] some interfaces are evaluated according to the requirements defined in their specific specification:

- DMI Interface: ERA/ERTMS/015560 [4]
- JRU Interface: SUBSET-027 [5]
- TIU Interface: SUBSET-034 [6]

3.4.2 Evaluation of Dependent and Independent Steps in Case of Special Step Results

This section defines the validation criteria and continuation rules for the test when some steps are evaluated as Non-Passed, Optional Functionality, or Not Applicable.

3.4.2.1 Definition of Dependent and Independent Steps

A step is considered as dependent when its execution or validation relies on the occurrence of a previous step.
Here are some examples: the recording of an event in the JRU depends on a trigger from another interface, the validation of data entered by the driver typically depends on a preceding prompt requesting data input.

A step is considered as independent when its evaluation is based only on the time and/or distance defined in the test, and not on any previous events

3.4.2.2 Evaluation Criteria for Dependent steps

When two steps (Step A and Step B) are dependent, i.e. step B is a consequence of step A in the test, the validation of Step B depends on whether it is linked to Step A. Then, the following evaluation table shall be applied:

Step A Result	Step B Depends on Step A	Step B Does Not Depend on Step A
Non-Passed (no input found)	Step B = Non-Passed	Step B is evaluated independently
Non-Passed (input outside Validation ranges)	Step B = Linked to Previous Error	Step B is evaluated independently
Optional Functionality	Step B = Optional Functionality	Step B is evaluated independently
Not Applicable	Step B = Not Applicable	Step B is evaluated independently

Table 2: Dependent and independent steps

Note: Non-Passed – No Input Found: The expected input was not received by the system under test at all. Non-Passed – Input Outside Range: The expected input was received, but not within the validation ranges defined in the test specification.

This classification system ensures consistent, traceable evaluation and helps identify source errors versus inherited or dependent failures in the sequence.

4 GENERIC TYPES OF CHECKS

4.1 EVENT-BASED CHECKS

Event-based checks are checks following an event which is supposed to happen only once at a specific distance and time during the test. It can be applied to input and output steps.

4.1.1 Concept of validation ranges

In the test result, time and distances ranges shall be evaluated to verify that they are respected compared to the test specification. These ranges are used to assess that a specific test step is successfully passed or not. Time and distance ranges evaluation also allow to check that the test facility is properly stimulating the system under test in accordance with the designed sequence of test steps.

4.1.2 Time ranges

Considering that the train is at a distance specified in the step in a certain time, the validation of the time ranges shall consider an interval before and after that time.

This time interval shall be defined for each interface of the system under test depending also on the data flow, either input or output.

Given that the test facility is made up of different recording systems working with temporal data, all the timestamps of the entries used for evaluation shall be synchronized to the reference time selected.

4.1.3 Distance ranges

Each test sequence shall have a specified distance for each step, that must be checked. The validation of the distance ranges may consider an interval before and after that distance.

Considering the special case of standstill, also the time ranges shall be taken into account for the test step evaluation.

As an example, in an autonomous driving test, a possible check can be done verifying if the train stopped within a certain distance window.

4.2 CONTINUOUS CHECKS

Continuous checks shall be performed repeatedly throughout the entire test, without limitations on time or distance, with or without additional conditions based on other outputs (for example: outputs containing a validity bit, or an equivalent attribute, should only be checked when the system under test reports them as valid).

These checks may involve a direct examination and control of the system's outputs, which can either be a straightforward comparison of values or a more complex evaluation that considers multiple outputs and inputs. Depending on the context, it may be necessary to cross-reference various variables to assess the overall behavior of the system. Typically, such checks are more common (though not exclusive) in validation or verification tests.

For instance, if the system provides real-time data that must be continuously monitored, such as train localization information, it is crucial that the measured value (e.g., the position of the train) consistently matches the expected parameters, regardless of the changing test conditions. Therefore, the check is not limited to a one-time snapshot of the data but must track the system's behavior over time. This might include, for example, ensuring that the check of the confidence interval for train localization always contains the actual position of the train (provided by the test bench) throughout the test.

While these types of checks are primarily used in validation, they can also extend to other types of testing, such as performance tests, where the focus is not only on the correctness of the results but also on quantitative parameters that assess the system's overall performance. In such cases, the analysis might include statistical calculations, such as the integrity rate of the localization or the average size of the confidence interval, to ensure that these values remain within predefined limits.

In summary, these checks shall be performed continuously, reflecting the system's behavior throughout the steps, and ensuring that all operational conditions are constantly monitored to verify the correct functioning of the system.

4.3 CHECKS ON TIME/DISTANCE WINDOWS

Checks on time/distance windows are similar to those described in §4.2. These time or distance windows on which these tests are focused shall be defined depending on the objectives of the test. For example, a window might be defined as the period between two specific distances travelled by the train, or it might be defined as a time interval within which certain conditions are met. Alternatively, checks could be limited to a specific time duration following a particular event, such as an anomaly detection, or a trigger point.

Moreover, these checks can be applied not only to individual time/distance windows but also to a sequence of successive windows. In such cases, the system's performance or behavior is monitored across several contiguous intervals, and the checks are performed on the concatenation of these windows, ensuring that the system behaves correctly throughout the entire duration of the test, even if the conditions change between windows.

This approach allows for more focused assessments within specific segments of the test, rather than requiring continuous monitoring throughout the entire duration of the test. It enables more targeted

validation of the system's behavior under defined circumstances or during critical phases of operation.

For example, in a train localization function validation, a possible check can be done when the train is traveling between two specific points, or during a fixed time window after the train has passed a certain checkpoint. These time/distance windows could also be used to analyze the system's response to specific events, such as acceleration, deceleration, or stops, allowing for more granular validation of its performance under different operational conditions.

5 EXAMPLES OF APPLICATION AND ADAPTATION OF EVALUATION RULES FOR SOME INTERFACES

Depending on the subsystems of the CCS+ system under test, the variety of checks needed for verification/validation or performance tests differs.

For instance, for ETCS-OB, the certification tests defined by the SUBSET-076 [7] only rely on event-based checks.

In the following sections, it is proposed to focus on the Advanced Safe Train Positioning (ASTP), Automatic Train Operation (ATO), Digital Register and Onboard Communication Network since several deliverables of WP34 already mentioned these subsystems.

5.1 ADVANCED SAFE TRAIN POSITIONING

5.1.1 Verification and Validation tests

5.1.1.1 Continuous checks

Some verification and validation checks specific to the ASTP that have been identified so far belong to the category of continuous checks, introduced in 4.2. These checks aim at verifying the behaviour of the system under test through the entire test.

For instance, continuous checks seem particularly relevant for tests related to safety.

Regarding the ASTP, the most obvious example is that when provided by the subsystem - i.e. available in the outputs of the ASTP (which must be confirmed by a validity bit) - the confidence interval for the train localization must "always" (from a requirement point of view: within the most constraining THR) contain the real localization of the train which is simulated by the test bench. Requirement FP2-ASTP-SRS-056 specifies this expected behaviour in D21.2 – System requirements of ASTP system [3]. The rationale is that a real position of ASTP reference point not included in the confidence interval can lead to catastrophic events (collision or catch-up for instance).

Other examples are:

- When provided by the subsystem, the track ID associated to the train localization must "always" (from a requirement point of view: within the most constraining THR) match a track included in the simulated train route (requirement FP2-ASTP-SRS-068 in [3]). The track ID provided by the ASTP may be different from the one on which the estimated

position of the train is currently located: the feared event is the provision of a track ID not included in the real train route (wrong track after crossing a switch for instance).

- When provided by the subsystem, the confidence interval for the train speed must “always” (from a requirement point of view: within the most constraining THR) contain the real speed of the train which is simulated by the test bench (requirement FP2-ASTP-SRS-067 in [3]). The rationale is that a real train speed exceeding the max safe speed provided by the ASTP can lead to catastrophic events (derailment for instance).
- When provided by the subsystem, the confidence interval for the train acceleration must “always” (from a requirement point of view: within the most constraining THR) contain the real acceleration of the train which is simulated by the test bench (requirement FP2-ASTP-SRS-068 in [3]). The rationale is that a real train acceleration not included in the confidence interval can lead to catastrophic events (collision or catch-up due to erroneous braking curves calculation for instance).

Analysis tools (evaluation in post-treatment and display in real time during the test) should also propose a graphical representation (in addition to the official report) to highlight the possible defaults regarding these continuous checks.

5.1.1.2 Event-based checks

Some verification and validation checks specific to the ASTP that have been identified so far belong to the category of event-based checks, introduced in 4.1. These checks aim at verifying the behaviour of the system under test after a specific event occurs (or at a specific location).

Regarding the ASTP, one example may be to check that after the reception of the message from a balise group installed for localization purpose, the confidence interval for the train localization is reset, the estimated localization of the train is relocated (if needed) and the selection of the track is confirmed (if not confirmed before).

Another example is that in case of malfunction (sensor failure or unavailability), if the ASTP is not able to produce a data respecting the safety requirements (in terms of THR only for the data related to safety), it must be checked that the ASTP provides default invalid values for the related data and a function status set to “Dataset unavailable” for this data but keep providing all other data as specified (requirements FP2-ASTP-SRS-013 and FP2-ASTP-SRS-083 in [3]). For instance, if the ASTP cannot provide a SIL4 confidence interval, it shall be set to a default invalid value but the ASTP may provide an estimated position that can be used by other subsystems which do not need a safe position.

More specifically, at the end of initialization of the ASTP, if the ASTP cannot reach full operational capability for any reason (weak GNSS signal…), it can be checked that estimated speed and travelled distance (not safety related data) since the ASTP is powered on are always provided anyway (requirement FP2-ASTP-SRS-020 in [3]).

5.1.2 Performance tests

5.1.2.1 Checks on time/distance windows

Some performance checks specific to the ASTP that have been identified so far belong to the category of checks on time/distance windows, introduced in 4.3. These checks aim at verifying the

behaviour of the system under test in specific areas, which can be defined with a starting and ending point (using time or distance).

Regarding the ASTP, one example of relevant performance check on distance windows is the monitoring of the mean or maximum value of the size of the confidence interval for the localization in specific areas, like:

- tunnel areas,
- dense urban environment areas,
- open sky areas,
- mountain environment areas…

In these specific areas, the expectations regarding the performances of the ASTP may differ so this is why a distinction can be made.

According to [3] (requirements FP2-ASTP-SRS-036 and FP2-ASTP-SRS-037), the distinction could rather be made for areas “with constraints if the train is stopping in this area” (the maximum computed ½ confidence interval is 10m in these areas, whereas in other areas it is 60m).

An example of performance check on time window is the monitoring of the initialization of the ASTP.

According to [3] (requirements FP2-ASTP-SRS-018 and FP2-ASTP-SRS-019), the ASTP shall be fully operational in less than 1 or 5 minutes (depending on the status of initial position and track ID – saved or not before ASTP was switched-off – and information received from CMD).

This kind of requirements can easily be checked with a time window starting when the ASTP is powered-on.

5.1.2.2 Continuous checks

For the ASTP, several performance checks belong to the category of continuous checks.

According to [3]:

- The computed safe ½ confidence interval towards estimated speed shall not exceed 2 km/h for speeds lower than 30 km/h and not increase linearly up to more than 12 km/h for speeds between 30 km/h and 500 km/h (requirement FP2-ASTP-SRS-080).
- The computed safe ½ confidence interval towards estimated acceleration shall not exceed 0.2 m/s^2 (requirement FP2-ASTP-SRS-041).
- The safe maximum distance (Estimated distance max) and the safe minimum distance (Estimated distance min) under and over estimation compared to the estimated distance travelled (Estimated distance travelled) shall be lower than 2% of the travelled distance (requirement FP2-ASTP-SRS-049).

These performance requirements must be checked through the entire test, whatever the conditions of the test (nominal or degraded).

Other requirements specified in [3] are dealing with availability rates or performance expectations that must be met in “xx% of cases”, like:

- FP2-ASTP-SRS-051: The ASTP shall have an overall availability of 99.998% during operation.

- FP2-ASTP-SRS-038: For at least 95% of the cases, the absolute error of the ASTP reference estimated position provided by ASTP (difference between the real position of the train simulated by the test bench and the estimated position provided by ASTP) shall not exceed +/-5m.
- FP2-ASTP-SRS-040: For at least 95% of the cases, the absolute error of the estimated speed provided by ASTP (difference between the real speed of the train simulated by the test bench and the estimated speed provided by ASTP) shall not exceed +/- 1 km/h for speeds from 0 km/h to 100 km/h and +/- 1% * v for speeds from 100 km/h to 500 km/h.
- FP2-ASTP-SRS-043: For at least 95% of the cases, the absolute error of the estimated acceleration provided by ASTP (difference between the real acceleration of the train simulated by the test bench and the estimated acceleration provided by ASTP) shall not exceed 0.05 m/s².
- FP2-ASTP-SRS-044: For at least 95% of the cases, the absolute error of the estimated 3D ASTP reference position shall not exceed 2.5m on the x y and z axis of the carriage reference frame.
- FP2-ASTP-SRS-045: For at least 95% of the cases, the absolute error of the estimated 3D train velocity shall not exceed 2 km/h on each axis of the carriage reference frame.
- FP2-ASTP-SRS-046: For at least 95% of the cases, the absolute error of the estimated 3D train acceleration shall not exceed 0.05 m/s² on each axis of the carriage reference frame.
- FP2-ASTP-SRS-048: For at least 95% of the cases, the absolute error of the estimated 3D attitude (rotational angles) shall not exceed 0.1° for yaw, and 0.5° for pitch and roll angles.

To check these requirements, tests will be defined with different external and environmental conditions (good/nominal, intermediate, bad/degraded…).

On the contrary, performance requirements listed above are specified with figures that apply overall, considering all the operational situations that the ASTP will encounter in its lifecycle.

It is very likely that for testing, the performances objectives and evaluation criteria must be declined for each test according to the type of conditions simulated, in such a way that overall, the initial performance requirement specified in [3] is met. For instance, tests in degraded conditions for FP2-ASTP-SRS-051 can be designed with an expected availability rate as evaluation criterion which is lower than the overall expected availability rate specified by FP2-ASTP-SRS-051. On the contrary, expected availability rate in tests with nominal conditions should be greater than the overall expected availability rate.

Therefore, one of the main challenges here is to define evaluation criteria that depend on the test conditions and that must be carefully chosen to meet the overall evaluation criterion specified by the performance requirement. The way to define the evaluation criteria for each type of test conditions remains to be decided and may differ from one subsystem to another.

For ASTP, the overall performance criteria described in the requirements listed above are only examples and are still subject to evolution, especially considering the results of the demonstrations done in the scope of WP22. Task 22.5 will elaborate KPIs to evaluate and assess the performance

of the demonstrators. The outcome of this study will help to better understand what is achievable by the ASTP and will help to specify the performance requirements. These KPIs will address scenarios with different external and environmental conditions in order to assess the overall performances of the ASTP. Results obtained with these KPIs for each set of specific external conditions could be used to define the evaluation criterion of the associated test (with the same conditions) for the ASTP.

For this kind of requirements, the check is considered as continuous because the behaviour of the system under test during the entire test has an influence on the result. For instance, for FP2-ASTP-SRS-051, the availability of the ASTP during the entire test is computed in post-treatment and must be compared to the expected results for the particular test conditions. This statement is also true for the other requirements listed above with performance expectations that must be met in “xx% of cases”.

5.2 AUTOMATIC TRAIN OPERATION UP TO GOA2

SUBSET-125 [10] specifies the system requirements for the ATO-OB and ATO-TS which are considered as interoperability constituents in the CCS TSI 2023 [11].

The objectives of the implementation of ATO-OB and ATO-TS are to improve capacity, on-time performance and energy efficiency by driving the train automatically and adapting the speed to different constraints.

The functionalities are limited to GoA2: the train is driven automatically (automatic stop) but a driver is still present in the cab to perform some manual functions (departure of the train, doors operation depending on the train configuration…).

Several WPs of R2DATO (including WP6) are working on ATO GoA3-4 (driverless/unattended train operation) specifications but this section only focuses on ATO GoA2 as specified by [10].

Some examples of application of evaluation rules for the ATO-OB and ATO-TS are given in the following sections. This exercise is not exhaustive and all the system requirements that are referenced are specified in [10]. The definitions of some terms specific to ATO that are used in the following sections can be found in the ERTMS/ATO Glossary [12].

5.2.1 Verification and Validation tests

5.2.1.1 Event-based checks

Some verification and validation checks specific to the ATO that have been identified so far belong to the category of event-based checks, introduced in 4.1.

ATO-OB

Regarding the ATO-OB, there are a lot of examples of event-based checks in verification and validation tests. Some actions from the driver or inputs from the rolling stock have direct effects on the expected ATO-OB behavior.

For instance:

- the ATO-OB shall disengage if the driver activates manually the brake via the Traction/Brake Lever (SUBSET-125-6.2.1.11),
- the ATO-OB shall remain engaged if the driver commands traction manually via the Traction/Brake Lever (SUBSET-125-6.2.1.12),
- the ATO-OB shall delete any stored Journey Profile and it shall send a new Handshake Request if the Train Running Number changes (SUBSET-125-6.4.2.9),
- the ATO-OB shall send an automatic door closing command to the train door management system when the remaining dwell time is equal to the train configuration time to perform normal door closure procedure (if the static train configuration allows automatic train doors closing and the ETCS-OB is in AD mode, SUBSET-125-7.2.2.6),
- the ATO-OB shall consider the next Stopping Point as a Stopping Point to be skipped if the driver requests to skip it (SUBSET-125-7.3.1.6),
- the ATO-OB shall send the low adhesion information to the ATO-TS when low adhesion is selected by the driver (SUBSET-125-7.5.1.1),
- the ATO-OB shall send the “slip/slide” information to the ATO-TS when it is informed about “slip/slide” by an external system (SUBSET-125-7.5.1.2),
- all the ATO-OB state transitions (SUBSET-125-9.10.6).

ATO-TS

Regarding the ATO-TS, the following examples can be given as event-based checks in verification and validation tests:

- the ATO-TS shall request Train Hold at a Stopping Point by means of a Journey Profile update when some conditions are met (SUBSET-125-7.4.1.1),
- on setting up a communication session with the ATO-OB, after having received the ATO system version(s) supported by the ATO-OB, the ATO-TS shall answer with the ATO system version that will be used by the ATO-TS for the ATO-OB/ATO-TS communication (SUBSET-125-7.9.3.2).

5.2.1.2 Checks on time/distance windows

Some verification and validation checks specific to the ATO that have been identified so far belong to the category of checks on time/distance windows, introduced in 4.3.

ATO-OB

Regarding the ATO-OB, there are also a lot of examples of checks on time/distance windows in verification and validation tests. Some examples are given below:

- the ATO-OB shall drive the train automatically while it is engaged (which can be assimilated to a distance window, SUBSET-125-6.2.1.1),
- the ATO-OB shall maintain the train stationary requesting to the train the application of the Train Holding Brake (SUBSET-125-6.2.1.7),
- the ATO-OB shall drive the train at a speed lower than or equal to the ETCS permitted speed received from the ETCS-OB when the ETCS-OB is in ceiling speed monitoring (which can be assimilated to a distance window, SUBSET-125-7.1.3.2),
- the ATO-OB shall drive the train so as not to reach the ETCS EBI intervention limit received from the ETCS-OB when the ETCS-OB is in target speed monitoring (which can be assimilated to a distance window, SUBSET-125-7.1.3.3),
- the ATO-OB shall drive the train at a speed lower than or equal to the ETCS release speed received from the ETCS-OB when the ETCS-OB is in release speed monitoring (which can be assimilated to a distance window, SUBSET-125-7.1.3.6),
- the ATO-OB shall take into account the following information (that can be assimilated to distance windows) from the Journey Profile and Segment Profiles affecting the Traction or Brake effort limits in order to compute the traction/braking commands: Low adhesion areas, Current consumption limitation zones, Switch off Regenerative Brake areas, Switch off eddy current brake for service brake areas, Switch off eddy current brake for emergency brake areas and Switch off Magnetic Shoe Brake areas (SUBSET-125-7.1.5.2).

ATO-TS

Regarding the ATO-TS, the following example can be given as check on time/distance windows in verification and validation tests:

- the ATO-TS shall inform the ATO-OB using the Segment Profiles or Journey Profiles, if it is not possible to apply traction, e.g. due to passing through a powerless section which can be assimilated to a distance window (SUBSET-125-6.2.1.14).

5.2.2 Performance tests

5.2.2.1 Event-based checks

Some performance checks specific to the ATO that have been identified so far belong to the category of event-based checks, introduced in 4.1.

ATO-OB

Regarding the ATO-OB, the following examples can be given as event-based checks in performance tests:

- the ATO-OB shall be able store at least two maximum size Journey Profiles including the maximum number of referenced maximum size Segment Profiles (the event corresponding to the reception of this information, SUBSET-125-6.4.2.15),
- the ATO-OB is required to stop the train at a position in rear of an EOA, as specified in the Segment Profile (SUBSET-125-7.1.3.9).

5.2.2.2 Continuous checks

Some performance checks specific to the ATO that have been identified so far belong to the category of continuous checks, introduced in 4.2.

ATO-OB and ATO-TS

Regarding the ATO-OB and ATO-TS, the following example can be given as continuous check in performance tests:

- the ATO-OB and ATO-TS shall use the UTC Time with an accuracy of ±1 second (SUBSET-125-7.6.1.1).

5.3 DIGITAL REGISTER

R2DATO WP27 oversees the specification, development and implementation of the Digital Register (DR). The scope of the DR is similar to the concept of Digital Map from projects like RCA, OCORA, and S2R.

The DR is a data management system that processes data from existing railway databases or issued from engineering processes and makes this data available to consuming systems or subsystems with centralized and standardized distribution processes. The DR is also responsible for managing the safe activation of new data versions within the consuming systems or subsystems. The DR only handles static (data that does not change as long as the data version is valid) infrastructure data (i.e., track layout incl. geometry, track objects, track civil structures). The consuming systems or subsystems can use this data for safety-related and non-safety-related use cases.

DR consists of a trackside (DR-TS) and on-board (DR-OB) subsystem.

DR-TS receives, aggregates and validates engineering data from the IM, and then compiles it to Domain Data that it distributes to trackside Consuming Systems (for instance TMS, MBS and ATO-AT). It also sends the Domain Data to the DR-OB via an airgap interface.

DR-OB is part of the Repository On-Board, which is an on-board repository to store Domain Data in the train. The DR contributes to the data collection of this repository by providing the required Domain Data (or Map Data in the context of on-board) to on-board Consuming Systems (like ASTP). The Repository On-Board also stores other types of data: Journey profiles (operational data), Train Data and Mission Data (received from the RU Control Center). It is not clear yet what will be the modularity chosen in the on-board reference architecture: will the subsystem to consider for certification be the DR-OB or the whole Repository On-Board? For the moment, this deliverable is focusing on the DR-OB (scope of WP27).

Currently, the deliverable D27.1 – Set of requirements on the Digital Register in R2DATO [13] only presents an initial set of functional requirements for the DR and an overview of the interfaces from a logical point of view to ease the understanding. Detailed interface requirements, non-functional requirements (e.g. RAMSS), technical requirements and data conversion requirements will be defined later (deliverable D27.3 plans to provide an updated set of requirements).

The deliverable D27.2 – Specification of Digital Register Implementation(s) required in R2DATO [14] describes more deeply the logical architecture of the DR as well as, but only for the trackside part (DR-TS), exchange scenarios and interfaces with more details (logical definition but also messages structures).

Some examples of application of evaluation rules for the DR-OB and DR-TS are given in the following sections. This exercise is not exhaustive and all the system requirements that are referenced are specified in [13] (more information regarding the functionalities of the DR can also be found in this document).

For the time being, the specifications currently available only allow us to identify potential verification and validation tests (no performance tests).

5.3.1 DR-TS

5.3.1.1 Event-based checks

Regarding the DR-TS, there are a lot of examples of event-based checks in verification and validation tests. Some actions from external actors have direct effects on the expected DR-TS behavior.

For instance:

- when an Import Trigger is received from DR Data Manager, the DR-TS shall aggregate Engineering Input Data received from one or multiple external source(s) into a common set of Engineering Data (SysF2),
- when the check of the consistency of Engineering Data is not successful, the DR-TS shall report any validation Issue (data issues or issues during validation) to the DR Data Manager (SysF3),
- when the check of the consistency of Engineering Data is successful, the DR-TS shall start the compilation and transform Engineering Data into the Consuming System specific Domain Data (SysF5),
- when the verification of the resulting Domain Data regarding defined rules and data definitions is not successful, the DR-TS shall report any Compilation Issue (data issues or issues during processing) to the DR Data Manager (SysF5),
- when the verification of the resulting Domain Data regarding defined rules and data definitions is successful, the DR-TS shall report the Compile State to the DR Data Manager (SysF5),
- the DR-TS shall update the existing Domain Data version with the newly generated Domain Data (SysF8),

- the DR-TS shall distribute Domain Data based on the received Domain Data Activation Time to the Consuming System (SysF8),
- when a Consuming System requires an update of Domain Data, the DR-TS shall transact the current active version of Domain Data to the Consuming System (SysF9),
- when the Authorisation command is received from DR Data Manager as well as completed Domain Data distribution process, the DR-TS shall transact the Domain Data Activation Request to TMS (SysF12),
- when the Domain Data usage restriction acknowledgement is received from MBS, the DR-TS shall authorise the change in required version of Domain Data for the consuming systems and shall send the activation commands for the new Domain Data version to all Consuming Systems (SysF13),
- when the Activation acknowledgements are received from all the consuming systems, the DR-TS shall send Activation commit status to all the consuming systems to commit the activation process (SysF13).

5.3.1.2 Checks on time/distance windows

Regarding the DR-TS, there are also a few examples of checks on time windows in verification and validation tests. Some examples are given below:

- the DR-TS shall monitor the loading of Domain Data from the Consuming Systems (SysF8),
- the DR-TS shall monitor the incoming activation acknowledgements (SysF13).

5.3.2 DR-OB

5.3.2.1 Continuous checks

Some verification and validation checks specific to the DR-OB that have been identified so far belong to the category of continuous checks, introduced in 4.2:

- the DR-OB shall perform a Map relevancy check between the available minimum/maximum required Map Reference Data in the DR-OB and required minimum/maximum required Map Area based on current train position to ensure completeness/availability of Map Data in train (SysF14),
- the DR_OB shall ensure that consuming systems in train always have the current active version of Map Data (SysF14),
- the DR-OB shall always check that the current available Map Data in the DR-OB corresponds to at least the minimum required map area (SysF14).

5.3.2.2 Event-based checks

Regarding the DR-OB, there are also a lot of examples of event-based checks in verification and validation tests.

For instance:

- when the Required Map Area from ASTP AND Minimum and Maximum required Map Reference Data does not correspond to Map Data in the DR-OB, the DR-OB shall classify the Map Data to be incomplete, send request for Map Data using Map Reference Data from the DR-TS and send deactivation command to consuming systems (SysF14),
- when a Map Reference Data is received from the DR-TS, the DR-OB shall determine the required update of Map Data version based on the available Map Reference Data version in the DR-OB and when it is complete, the DR-OB shall transact the valid version of Map Data to consuming systems (SysF16),
- when the Map Data corresponds to the Map Reference Data AND integrity of Map Data is confirmed, the DR-OB classify the Map Data as valid and start the on-board activation process for the received Map Data, otherwise the DR-OB shall classify the Map Data as invalid and request the Map Data again from the DR-TS (SysF17),
- when a deactivation trigger is received or if the connection between DR-TS and DR-OB is lost, the DR-OB shall deactivate the Map Data in the DR-OB (SysF19).

5.4 ONBOARD COMMUNICATION NETWORK

R2DATO WP23 oversees the specification of system requirements for the Onboard Communication Network, based on a comprehensive set of user stories, in the deliverable D23.2 – Definitive and aligned requirements set for the Onboard Communication Network and Basic Services [15]. Then, it will explore possible solutions for these system requirements: these solutions are not available for the moment. Later, WP24 will focus on the network technical management functionality and associated processes. Currently, the WP34 can only make assumptions regarding the technical implementation and this deliverable does not go into technical details.

The Onboard Communication Network allows on-board building blocks to be interfaced and communicate together when they are not integrated in the same Computing Unit and it also allows the interaction between the on-board building blocks and the TCMS.

Some examples of application of evaluation rules for the Onboard Communication Network are given below. This exercise is not exhaustive and all the system requirements that are referenced are specified in [15].

The focus is made on non-functional system requirements as the functional system requirements currently specified in [15] are too high-level to apply evaluation rules.

Therefore, examples of checks are only given below for performance tests.

The performance tests that have been identified so far will more likely be associated to continuous checks:

- clock synchronization between end devices shall always have an accuracy of 1ms (ComStackReq-NonFunc-29 and ComStackReq- NonFunc-32),
- for process data, the latency shall be limited to 10 ms (ComStackReq-NonFunc-30),

- for spontaneous/event-driven communication, the latency shall be limited to 100 ms (ComStackReq-NonFunc-31),
- on all shared links, the supported link speed shall be at least 1 GBit/s (ComStackReq-NonFunc-34),
- for end devices, the supported link speed shall be at least 100 MBit/s, full and half duplex (ComStackReq- NonFunc-35).

6 TEST REPORT

6.1 CONTENT

The test report is a document that collects and describes the results of the testing activities executed on the system under analysis. The document must have different sections that contain the main information related to the activity performed:

1) **Introduction:** This section contains a description of the purpose of the document, a summary description of the content of each chapter, the list of acronyms and the table of referenced documents. The table of referenced documents must certainly include the Test Specification documentation, the documentation to identify the version of the system under analysis, the documentation to identify the test environment used and the list of regulations and standards applied for the execution of the tests.

2) **Description of the system under test:** In this section it is necessary to identify the system under test, clearly defining the versions of the various subsystems that constitute it and referencing the associated documentation for each (Version Description Document). When it is permitted, in a single test report different versions of the system under test are listed, each with the detailed identification of the corresponding subsystems; this happens when some tests have been executed using the latest version of the system under test while others refer to old executions performed with previous versions of the system under test that do not require re-execution with the final version. The use of different version in a report is not allowed for official Subset-076 certification test campaigns. This section also contains a short textual description of the system under test describing the main features.

3) **Test environment description:** This section contains a short description of the Hardware and Software tools used to provide input to the system under test. In addition, the test environment is also used to store the logs of the system under test and to analyze them automatically. It describes how these tools interact with each other, how the tests are executed (manually, automatically, using scripts...) and how the input information is sent to the system under test. The tools used to define the scripts needed to execute the tests must also be described. For each tool used to define or execute the tests it is necessary to perform a unique identification indicating the version and the reference documentation.

4) **Summary of results:** in this section the results of the execution of the tests are presented in table form; for each test it is necessary to indicate:
    a. ***Name of the test;***
    b. ***Reference to the test specification;***
    c. ***Version of the System under test;***

*d. **Version of the test execution environment;***

*e. **Execution date (optional);***

*f. **Name of the logs;***

g. ***Test Result:*** refer to the instructions given in chapter 3.3.2.

h. ***Related WI/SPRs:*** in the event that the test is not passed the problem identified must be recorded in a Work Item or Software Problem Report; in this section, the unique identifiers of the WI/SPRs created for the not passed test must be indicated;

i. ***Notes:*** In this field detailed information is indicated relating to the individual test; in particular, if the test is referenced to an old execution performed with a version of the system under test prior to the final version it is necessary to indicate why this execution is also considered applicable to the final version of the system under test.

5) **Detailed results:** in this section of the test report, for each test, a copy of the test specification is reported, with all the input/output steps; for each step, a column is added relating to the outcome of the execution according to the instructions given in chapter 3.3.1. In case a test step is actually modified respect to the original one in the Test Specification or partially executed or not executed, the following information have to be provided:

- a justification for the change for the test step
- a justification of no impact on the coverage of the requirements under test.
- a justification for partial/no execution

In case an output step is Not Passed it is needed to indicate the relative WI/SPR id in the column relating to the execution outcome. In order to enhance the readability of the report and to limit the number of pages, the detailed evaluation results can be provided as an attached zip file or as Appendix to the report.

6) **Conclusions:** this section of the report indicates the total number of tests performed with the various versions of the system under test, the total number of tests passed, non-passed, test error, the list of WI/SPR still open and related to the execution of the tests, with the list of requirements not implemented or wrongly implemented.

7) **Appendix:** in this section it is possible to insert the analysis of the differences between the previous versions of the system under analysis and the final version; this analysis is a support for the re-use of tests performed on previous versions, in order to demonstrate that the old executions are also applicable to the final version. This section also contains all the log files coming out from the executed tests

6.2 TEMPLATE

The Test Report shall be written filling the blank template [1].

7 REQUIREMENTS FOR ANALYSIS TOOLS

7.1 TOOLS RUNNING DURING THE TEST

The tools running in real time during the test execution must be compliant with the following requirements.

7.1.1 Functional requirements

REQ_ANALYSIS_TOOL_FUNC_01_01 The analysis tool shall display the current status of the test execution.

[Mandatory: Yes]

[Originated from REQ_FUNC_08_01 of [2]]

REQ_ANALYSIS_TOOL_FUNC_01_02 The analysis tool shall display all the relevant information allowing the tester to identify and detect anomalies as soon as they occur during test execution.

[Mandatory: Yes]

[Originated from REQ_FUNC_08_03 of [2]]

REQ_ANALYSIS_TOOL_FUNC_01_03 The analysis tool shall display all the relevant information (inputs/outputs of the system under test) allowing the tester to check the correctness of the current test execution.

[Mandatory: Yes]

[Originated from REQ_FUNC_08_04 of [2] and §5.1.1.1]

REQ_ANALYSIS_TOOL_FUNC_01_04 The analysis tool should display all the accessible and relevant internal data of the system under test allowing the tester to check the correctness of the current test execution.

[Mandatory: No]

[Originated from REQ_FUNC_08_04 of [2]]

REQ_ANALYSIS_TOOL_FUNC_01_05 The analysis tool shall generate logs and reports to allow the tester and other tools to perform offline analysis.

[Mandatory: Yes]

[Originated from REQ_FUNC_09_01 of [2]]

REQ_ANALYSIS_TOOL_FUNC_01_06 The analysis tool shall generate the logs and reports in a standardised format (CSV/Excel/JSON/SQL/HDF5).

[Mandatory: Yes]

[Originated from REQ_FUNC_09_02 of [2]]

REQ_ANALYSIS_TOOL_FUNC_01_07 The analysis tool shall collect the log files and juridical data of the system under test.

[Mandatory: Yes]

[Originated from REQ_FUNC_09_03 of [2]]

REQ_ANALYSIS_TOOL_FUNC_01_08 The analysis tool shall store the logs and reports in a database with easy access for offline analysis.

[Mandatory: Yes]

[Originated from REQ_FUNC_09_04 of [2]]

REQ_ANALYSIS_TOOL_FUNC_01_09 The analysis tool shall timestamp all the information stored in the logs and reports and ensure the synchronization of these timestamps based on a time reference.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_01_10 The analysis tool shall allow the tester to select the information to be stored in logs and report to limit it to the data useful for the test evaluation (in case several building blocks or subsystems have been integrated into the tests facilities).

[Mandatory: Yes]

REQ_ANALYSIS_TOOL_FUNC_01_11 If information displayed on a screen (e.g. DMI) must be checked for the test evaluation, the analysis tool shall record this information in a video file and store it in a database with easy access for offline analysis.

[Mandatory: Yes]

7.1.2 Quality requirements

REQ_ANALYSIS_TOOL_QUAL_01_01 The analysis tool shall be evaluated and validated before being used within the test environment.

[Mandatory: Yes]

[Originated from REQ_QUAL_02_02 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_02 The analysis tool should be compliant with the quality requirements specified by CENELEC EN 50716:2023 [9] or, if the development has started before end of October 2026, by EN 50128:2011/A2-2020 [8].

[Mandatory: No]

[Originated from REQ_QUAL_02_03 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_03 The analysis tool shall be easy to use.

[Mandatory: Yes]

[Originated from REQ_QUAL_04_01 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_04 The GUIs of the analysis tool shall be user-friendly.

[Mandatory: Yes]

[Originated from REQ_QUAL_04_02 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_05 The analysis tool shall be compatible with the cycle time(s) of the system under test, meaning that it is able to log and record all the outputs of the system under test.

[Mandatory: Yes]

[Originated from REQ_QUAL_05_02 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_06 Documentation regarding the use, set-up, and configurability of the analysis tool shall be available.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_01 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_07 All the modifications of the analysis tool (bug fixing, enhancements) shall be traced and available for consultation.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_02 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_08 The documentation related to the analysis tool shall be compliant with high standards of quality.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_03 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_09 The documentation shall be written in English.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_05 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_10 The documentation related to the analysis tool should be compliant with CENELEC EN 50716:2023 [9] or, if the development has started before end of October 2026, with EN 50128:2011/A2-2020 [8].

[Mandatory: No]

[Originated from REQ_QUAL_06_06 of [2]]

REQ_ANALYSIS_TOOL_QUAL_01_11 The analysis tool shall provide feedback and status report to supervise the test execution.

[Mandatory: Yes]

[Originated from REQ_QUAL_09_04 of [2]]

7.2 TOOLS RUNNING IN POST TREATMENT

The tools running in post treatment (after the test execution) must be compliant with the following requirements.

7.2.1 Functional requirements

REQ_ANALYSIS_TOOL_FUNC_02_01 The analysis tool shall automatically evaluate the results of test scenarios by checking the correctness of the expected behaviour (step-by-step).

[Mandatory: Yes]

[Originated from REQ_FUNC_09_05 of [2]]

REQ_ANALYSIS_TOOL_FUNC_02_02 If information displayed on a screen (e.g. DMI) must be checked for the test evaluation, the analysis tool shall be able to automatically analyse the video files to evaluate the results of test scenarios.

[Mandatory: Yes]

REQ_ANALYSIS_TOOL_FUNC_02_03 The analysis tool shall provide an overview about the automatic evaluations/results of executed tests, allowing easy analysis for the tester.

[Mandatory: Yes]

[Originated from REQ_FUNC_09_06 of [2]]

REQ_ANALYSIS_TOOL_FUNC_02_04 The analysis tool shall interface with a database containing the data stored during test execution (logs and reports) to allow the automatic evaluation to be performed.

[Mandatory: Yes]

[Originated from REQ_FUNC_14_02 of [2]]

REQ_ANALYSIS_TOOL_FUNC_02_05 The analysis tool shall allow the tester to manually access and browse the database containing the data stored during test execution (logs and reports) to allow the manual evaluation to be performed (in case of unsuccessful automatic evaluation or if there is any doubt regarding the automatic evaluation).

[Mandatory: Yes]

[Originated from REQ_FUNC_14_02 of [2]]

REQ_ANALYSIS_TOOL_FUNC_02_06 The analysis tool shall allow the tester to save notes and observations in the database.

[Mandatory: Yes]

[Originated from REQ_FUNC_14_03 of [2]]

REQ_ANALYSIS_TOOL_FUNC_02_07 The analysis tool shall automatically evaluate the test steps according to the available result categories (Passed, Passed with comments, Non-Passed, Linked to Previous Error, Optional Functionality, Not applicable, Test error, Observation and Deviation for Validation Tests; Passed and Non-Passed for Performance tests).

[Mandatory: Yes]

[Originated from §3.3]

REQ_ANALYSIS_TOOL_FUNC_02_08 The analysis tool shall be able to automatically evaluate the test steps according to event-based checks with a tolerance specified by time and distance ranges.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_02_09 The analysis tool shall be able to perform automatic continuous checks to monitor the system’s behaviour through the entire test.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_02_10 The analysis tool shall be able to perform automatic checks on distance windows (between two positions) to monitor the system's behaviour during some parts of the test.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_02_11 The analysis tool shall be able to perform automatic checks on time windows (delay after a specific event, time between two successive events or time during which specific conditions are met) to monitor the system's behaviour during some parts of the test.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_02_12 The analysis tool shall be able to perform automatic checks on successive time and/or distance windows.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_02_13 The analysis tool shall be able to perform automatic continuous checks and checks on time/distance windows based on the combination of several inputs/outputs, with possibly additional specific conditions.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_02_14 The analysis tool shall be able to perform automatic continuous checks and checks on time/distance windows based on statistical calculations.

[Mandatory: Yes]

[Originated from §4]

REQ_ANALYSIS_TOOL_FUNC_02_15 The analysis tool shall offer several ways to display the information contained in the database (logs and reports) to the tester and allow him/her to choose the most relevant one for the required manual analysis (graphical representation, tables…).

[Mandatory: Yes]

[Originated from §5.1.1.1]

REQ_ANALYSIS_TOOL_FUNC_02_16 The analysis tool shall allow the tester to manually and easily perform event-based checks, continuous checks and checks based on time/windows (with all the possibilities and constraints specified in REQ_ANALYSIS_TOOL_FUNC_02_08 to REQ_ANALYSIS_TOOL_FUNC_02_14; so with all the options regarding validation ranges, definition of time/distance windows, combination of data and calculations types like statistical ones).

[Mandatory: Yes]

REQ_ANALYSIS_TOOL_FUNC_01_12 The analysis tool shall provide tests results that can be graphically represented (tables, plots, diagrams) and navigated via GUIs.

[Mandatory: Yes]

REQ_ANALYSIS_TOOL_FUNC_01_13 The analysis tool shall provide a temporally aggregated logs view.

[Mandatory: Yes]

REQ_ANALYSIS_TOOL_FUNC_01_14 The analysis tool shall provide a verbosity filter when browsing logs.

[Mandatory: Yes]

REQ_ANALYSIS_TOOL_FUNC_01_15 The analysis tool shall provide a search function when browsing logs to highlight the requested information.

[Mandatory: Yes]

REQ_ANALYSIS_TOOL_FUNC_01_16 The analysis tool shall be capable of comparing log data and perform numerical elaboration on them to check correctness of results.

[Mandatory: Yes]

7.2.2 Quality requirements

REQ_ANALYSIS_TOOL_QUAL_02_01 The analysis tool shall be evaluated and validated before being used within the test environment.

[Mandatory: Yes]

[Originated from REQ_QUAL_02_02 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_02 The analysis tool should be compliant with the quality requirements specified by CENELEC EN 50716:2023 [9] or, if the development has started before end of October 2026, by EN 50128:2011/A2-2020 [8].

[Mandatory: No]

[Originated from REQ_QUAL_02_03 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_03 The analysis tool shall be easy to use.

[Mandatory: Yes]

[Originated from REQ_QUAL_04_01 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_04 The GUIs of the analysis tool shall be user-friendly.

[Mandatory: Yes]

[Originated from REQ_QUAL_04_02 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_05 The analysis tool shall provide tests results that are easy to access, read and analyse.

[Mandatory: Yes]

[Originated from REQ_QUAL_04_04 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_06 Documentation regarding the use, set-up, and configurability of the analysis tool shall be available.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_01 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_07 All the modifications of the analysis tool (bug fixing, enhancements) shall be traced and available for consultation.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_02 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_08 The documentation related to the analysis tool shall be compliant with high standards of quality.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_03 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_09 The documentation shall be written in English.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_05 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_10 The documentation related to the analysis tool should be compliant with CENELEC EN 50716:2023 [9] or, if the development has started before end of October 2026, with EN 50128:2011/A2-2020 [8].

[Mandatory: No]

[Originated from REQ_QUAL_06_06 of [2]]

REQ_ANALYSIS_TOOL_QUAL_02_11 The analysis tool shall generate test reports that are based on the template provided in the Annex [1] of this document.

[Mandatory: Yes]

[Originated from REQ_QUAL_06_04 of [2] and §6]

8 CONCLUSIONS

Based on the outcomes of the WP34 and learning from experience of other certification projects like the development of SUBSET-076 [7], generic principles for tests evaluation have been developed in the first sub-sections of this deliverable.

Generic rules for steps classification and evaluation have been defined. The different types of checks that must be supported by the analysis tools have also been documented.

These generic rules and statements may require some adjustments or particularities depending on the interfaces of the system under test that are being checked. This is why some examples of specific applications for the ASTP, ATO, Digital Register and Onboard Communication Network have been carried out. Lessons learned from these specific applications also allowed us to enrich and enhance the content of the generic parts of this document.

This deliverable also detailed the expected content for a certification test report and a template is available as an Annex of D35.3.

Finally, this deliverable has specified the requirements that analysis tools, needed for evaluation of the tests results, must fulfill. These requirements aim at summarizing all the generic principles that have been developed in the first sub-sections of this deliverable.

Distinction is made regarding the tools running in real time during the test execution and the tools running in post treatment. Objectives and associated constraints are different for these two types of tools. Therefore, the requirements are not the same for these both types of tools.

The generic principles for test evaluation and the associated requirements applying for analysis tools are likely to be enhanced and enriched again in the future. Feedback from other WPs is also expected to complete this first attempt of specifications for the analysis tools.

REFERENCES

[1] D35.3 Annex 1: Test Report Template.

[2] D34.3 – Tests facilities specifications, Revision 05

[3] D21.2 – System requirements of ASTP system, Revision 09

[4] ETCS Driver Machine Interface - ERA_ERTMS_015560 v4.0.0

[5] FIS Juridical Recording - SUBSET-027 v4.0.0

[6] Train Interface FIS - SUBSET-034 v4.0.0

[7] Test specifications – SUBSET-076 v4.0.0

[8] EN 50128 Railway Applications: Communications, signalling and processing systems – Software for railway control and protection systems. CENELEC European standard EN 50128:2011/A2-2020

[9] EN 50716 Railway Applications: Requirements for software development. CENELEC European standard EN 50716:2023

[10] ERTMS/ATO System Requirements Specification – SUBSET-125 v1.1.0

[11] Commission Implementing Regulation (EU) 2023/1695 of 10 August 2023 on the technical specification for interoperability relating to the control-command and signalling subsystems of the rail system in the European Union and repealing Regulation (EU) 2016/919

[12] ERTMS/ATO Glossary – 13E154, Version 2

[13] D27.1 – Set of requirements on the Digital Register in R2DATO, revision 05

[14] D27.2 – Specification of Digital Register Implementation(s) required in R2DATO, revision 03

[15] D23.2 Definitive and aligned requirements set for Onboard Communication Network and Basic Services, revision 04

[16] D34.1-General testing Strategy for a common virtual certification process

[17] D35.1-General testing Strategy for a common virtual certification process