OETCS/WP4/DescriptionOfWork

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

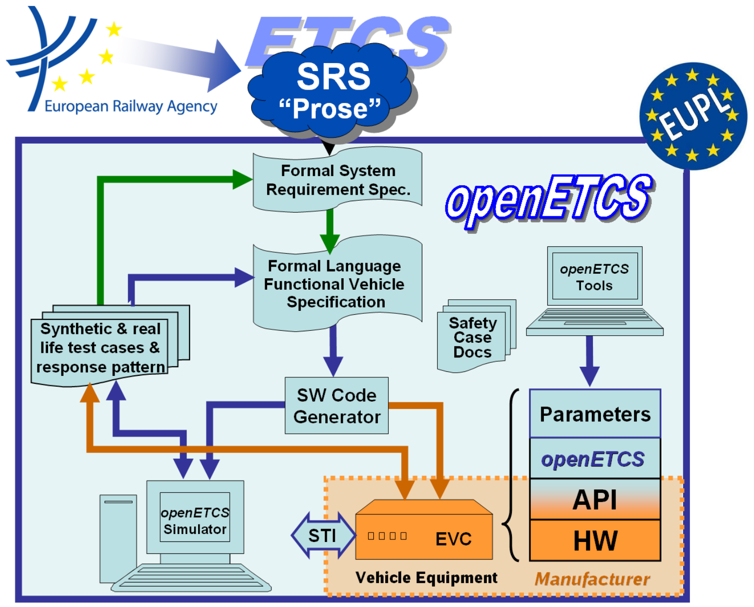
Work Package 4: “Validation & Verification Strategy”

openETCS Validation & Verification Strategy Work Package

Description of Work

Marc Behrens, Hardi Hungar, Ana Cavalli, Jens Gerlach, Hansjörg Manz, Jan Welte and Cyril Cornu

May 2013



Funded by:

Federal Ministry of Education and Research

Liberté • Égalité • Fraternité
République Française
Ministère de l'Enseignement Supérieur et de la Recherche

Investissements d'Avenir

Région de Bruxelles-Capitale

Gobierno de España
Ministerio de Industria, Energía y Turismo

This page is intentionally left blank

Work Package 4: “Validation & Verification Strategy” **OETCS/WP4/DescriptionOfWork**
May 2013

openETCS Validation & Verification Strategy Work Package

Description of Work

Marc Behrens

WP4 Leader

Hardi Hungar

WP4.1 Task Leader (Idetntification of tools and profile usage)

Ana Cavalli

WP4.2 Task Leader (Verification & Validation of the formal model)

Jens Gerlach

WP4.3 Task Leader (Verification & Validation of the implementation code)

Hansjörg Manz, Jan Welte

WP4.4 Task Leader (Verification of the tools and processes)

Cyril Cornu

WP4.5 Task Leader (Internal Assessment)

Description of work

Prepared for openETCS@ITEA2 Project

Abstract: This work package will comprise the activities concerned with verification and validation within openETCS. This includes verification & validation of development artifacts, that is, showing that models and code produced correctly express or implement what they are supposed to. And also, methods and tools to perform such tasks will be evaluated with the goal of assembling a suitable method and tool chain to support a full development.

The first output of this work package is a verification & validation strategy and plan. The plan identifies the verification & validation tasks to be done according to the openETCS life cycle (perhaps variants of the life cycle) chosen in WP 2. It proposes methods and means to perform the tasks and defines the orchestration of the single activities to a full verification & validation support of developing the EVC software.

Both model and code of the EVC software are subjected to verification & validation as they are produced by WP 3, and the tools and methods proposed in the verification & validation plan as well as those newly developed or improved in WP 7 are applied and evaluated in the process. Findings from these steps are iteratively fed back to the respective work package.

A dedicated activity studies the safety aspect of verification & validation. It takes into cosideration what the standards (mainly EN 50128 and EN 50129) mandate and defines how these requirements can be met by the combination of process, methods and tools.

The results of the work package are summarized in a final report on verification & validation at the end of the project.

Table of Contents

Figures and Tables

Figures

Tables

Introduction

Objectives

Verification is the activity to ascertain that a particular step in the development has achieved its goals, i.e., that its result correctly refines or implements its input, which may be a higher-level design or a specification. *Validation* is about making sure that the end result of the development meets its initial specification, that is, the requirements of the user. The term validation is also used when a design artifact is checked against requirements from previous steps. Verification and/or validation is required for most development artifacts. What exactly has to be checked depends on the set of items produced in the development process, their role and their nature. As the EVC software contributes to several safety-critical functions of the ETCS onboard unit, the specific requirements concerning the safety aspects of the standards EN 50128 and EN 50129 have to be respected throughout.

A main obligation of this work package is the verification or validation of development artifacts produced by WP 3. This work will concentrate on the functional and safety aspect. Besides verification & validation, the work package shall also establish a coherent and comprehensive chain of methods and tools for V&V in cooperation with WP 7. Specific challenges in this respect arise from the wish to use models extensively in the development process, which means that more common approaches have to be improved or substituted to fit a model-based development style, and from the requirement of using open-source tools, or even trying to realize the EVC software as an *open proof* item.

In pursuing these goals, the work package generates feedback concerning the adequacy, correctness and safety of development artifacts for WP 3, and the usefulness of tools and methods for WP 7.

Organisation of the Work package

The work packages consists of five tasks. The first task defines the verification & validation strategy and formulates the initial verification & validation plan. This plan defines the verification & validation activities to be done in openETCS and proposes the means to perform them. In later stages of the project the plan will be extended and revised to reflect the findings made while applying methods and tools to the artifacts at hand.

Both model and code of the EVC software are subjected to verification & validation as they are produced by WP 3, and the tools and methods proposed in the verification & validation plan as well as newly developed or improved tools from WP 7 are applied and evaluated in the process. Findings from these steps are iteratively fed back to the respective work package and used to refine the verification & validation plan.

A dedicated activity studies the safety aspect of verification & validation. It takes into consideration what the standards (mainly EN 50128 and EN 50129) mandate and defines how these requirements can be met by the combination of life cycle, methods and tools.

An internal assessment will simulate a real Assessor's task doing a Software Development assessment of the project impacting Working Packages 1, 2, 3, 4, 5, 6 and 7.

The phases defining the verification & validation process is divided into the design phase and the application phase. The *design phase* covers the time before the release of the artifacts which are to be evaluated. In this phase the findings of the last application phase are taken into account to improve verification & validation. The *application phase* covers the time after the release of the artifacts to be evaluated until the verification & validation report is written. For this phase the artifacts to be evaluated are frozen to a fixed release date.

Techniques for Verification & Validation

Verification & Validation techniques can be roughly classified into *dynamic* and *static* techniques. The most common dynamic verification & validation techniques are various forms of *testing*, which execute the code or the model. They are classified by their object or their purpose. These include:

- Unit testing
- Integration testing
- Acceptance test
- Software-in-the-Loop
- Model-in-the-Loop
- Model-based testing
- Monitoring
- Coverage analysis

A related dynamic activity is *animation*, which may play a role in analyzing an executable model.

Static verification & validation techniques—not executing model or code—include:

- Checking of coding guidelines
- Review
- Walkthrough
- Formal methods
  - Model checking
  - Deductive verification (theorem proving)
  - Abstract interpretation

%%description on V&V classification non formal-> formal -> formal -> code & description%%

Coping with a Model-Based Development Style

Models appear at different stages of the development. An important artifact of openETCS is a semi-formal model of the requirements. Depending on the modelling framework, the modelling language and formalization of the system requirements a concept has to be defined how the consistency and coherence of the model as well as the coverage of system requirements will be transparently verified. For this task, static verification techniques will very likely offer the best approach.

To verify that the model correctly captures the ETCS system requirement specification (Subset-026 et. al.), also dynamic techniques like animation might be useful. And finally, it may be helpful to also validate the model against the user requirements.

For later development stages, correct refinement or implementation of the model will have to be established. Again, techniques to be applied depend heavily on the nature of the model(s) and the process of how the code is derived. Model-based testing, i.e., deriving test cases from a model to ascertain that an executable behaves consistent to a model, is a technique to be used. Alternatively, if code is generated automatically from a model, other means like tools checking the correctness of a generation procedure (or its outcome on a case-by-case basis) may be chosen.

An important issue to be kept in mind is the suitability of models and tools for a safety-critical development. Modelling languages that lack a formal semantics or the expressive power to capture system aspects essential for safety considerations are of limited usefulness. And tools need to be qualifiable according to their role. For instance, a code generator needs to be verified or qualified or it must be accompanied by some tool checking the correctness of the generation step. Otherwise, the resulting code will have to verified similar as manually written code.

1 Identification of Tools and Profile Usage

The objective of this task is to prepare the activities of the tasks T4.2 and T4.3, which are concerned with verification & validation of the model and the code, respectively. It defines an overall verification & validation strategy and plans for verification and validation, detailing how and with what means the strategy is going to be implemented. Formally, the requirements on verification & validation which are to be covered in the plans are listed in D2.9 “Requirements for Verification & Validation”. An essential requirement is adhering to the applicable standards, mainly the EN 50128. The plans shall define activities adequate for a complete development, but also foresee a tailoring to the partial development actually realised in the project.

The WP 2 deliverable D2.3 “Process Definition” defines the development process and its steps, and thus also identifies the main verification steps. These will be detailed in the verification plan, defining what has to be achieved on a more technical level. Also, a selection of potentially applicable tools and methods will be given. The validation plan shall take the overall development approach including verification activities into account and define what methods are suitable for demonstrating that all requirements (to be defined by WP 2) are met by the end result, and also address the question of safety integrity.

Thus, the plans shall contain a selection of methods and a list tools suitable for applying the chosen V&V methods for[1]

1. the sub-system requirements specification (SSRS) and models (SFM, FFM)
2. the software semi-formal model and software architecture description
3. code derived from the software semi-formal model
4. the software strictly formal model and the software design description

[1] Terms according to the draft of D2.3, abbreviations according to the draft of D2.6.

D2.1 “Report on Existing Methodologies” shall already provide a list of potentially relevant methods and tools. Each method and tool applied in WP 4 shall be described in a format detailing its purpose, role and characteristics in terms of requirements on verification & validation. Not all steps will be automatic or semi-automatic. Manual techniques like review or walkthrough will play a role, too. In selecting tools, besides the requirement of openness (FLOSS), the question of qualification, depending on the role the tool will play, has to be answered. The formats describing methods and tools and criteria for their evaluation will be given in a document D4.1a “Preliminary Evaluation Criteria on Verification and Validation” supplementing the deliverable to be produced by this task, D4.1 “Report on V&V Plan & Methodology”.

Important classes of objects subjected to verification & validation activities are the different models, be they semi-formal or formal, the code derived from them and the versions of the demonstrator. Similar to methods and tools, these objects need to be defined w.r.t. their nature and role in the development process, including the requirements for verification & validation. A format for their description shall be provided with D4.1.

An analysis of these objects and the methods with which they are developed shall lead to a refinement and concretisation of the verification and validation plan in the course of the project after termination of T4.1. The plan is, as it is also common for any commercial standard-conforming system development, a living object which has to be revised and refined as the work proceeds. In the case of openETCS, the tasks T4.2 and T4.3 will do that after the termination of T4.1. The interim V&V reports appear as suitable steps with which this should happen.

Table 1. T4.1 Inputs, Outputs and Deliverables

T4.1 Identification of Tools and Profile Usage				
Type	Description	Due Date	Due Month	status
→	*D 2.1* Report on Existing Methodologies	Mar-2013	T0+9	no
→	*D 2.3* Process Definition	May-2013	T0+11	no
→	*D 2.4* Methods Definition	May-2013	T0+11	no
→	*D 2.9* Set of Requirements for V&V	May-2013	T0+11	no
D	*D 4.1a* Preliminary Evaluation Criteria on V&V	Mar-2013	T0+9	started
D	*D 4.1* Report on V&V Plan & Methodology	Jul-2013	T0+13	started

1.1 Creating the Backlog

Terminology

DAS2V: Design Artifact Subject to Verification or Validation

G: Goal

M: Means

F: Finding/Result

Goal: Completion of the V&V plan (in time)

WP4-T1-G: A useful plan for WP 4, that is, one that defines a way to achieve the goals of WP 4:

WP4-G1: Identify and demonstrate methods and tools to handle the V&V of a FLOSS development of the EVC software

WP4-G2: Perform as much of V&V on the DAS2Vs produced in the project as possible

Detailed Goals and Means

WP4-T1-G1: The plan shall give an overview of and a structure to the things required from V&V for an openETCS (FLOSS-) development.

WP4-T1-M1: Identifies all (most) of the activities which have to be made for a full development according to the standards, in a form relevant to the approach of openETCS (FLOSS, participants). This may include alternatives.

WP4-T1-G2: The plan shall provide a framework into which the V&V activities which will be performed within the project do fit.

WP4-T1-M2-1: Design formats for collecting information about DAS2Vs (V&V tasks), about the results of V&V activities, about activities of V&V method and tool development, about the results of evaluations of V&V methods and tools. Sketch how all of the information is to be gathered and finally incorporated into the final V&V report (D4.4).

WP4-T1-M2-2: Identify potential variants of partial implementations of V&V processes which are likely going to be performed within the project. These may be (?should be?) related to design activities within the project which produce DAS2Vs.

WP4-T1-G3: The plan shall delineate means for V&V within openETCS

WP4-T1-M3-1: A partial V&V process (see WP4-T1-M2 above) consists of a set of related DASVs and V&V steps to be applied to them. A V&V step is described by input and output (result, purpose) with V&V methods and means.

WP4-T1-M3-2: The plan will prepare the selection of adequate methods and means (tools) by providing evaluation criteria and incorporating available evaluation results.

WP4-T1-M3-2-1: Definition of an evaluation format for tools and methods.

WP4-T1-G4: The plan shall incorporate currently available information on openETCS development process and means and be amendable to future changes and additions.

WP4-T1-M4-1: Use D2.3 in instantiating the general requirements laid down in the standards.

WP4-T1-M4-2: Use D2.1 for tools.

WP4-T1-M4-3: Identify open points and include delineations for things which are useful for a complete V&V but not yet planned or detailed by project activities already performed.

Concrete First Steps (in SCRUM terminology: the backlog)

WP4-T1-S1: Assess the input material

WP4-T1-S1-1: Assess sketch of the V&V plan

WP4-T1-F1-1-1: The current format is .doc

WP4-T1-F1-1-2: The plan currently lists mainly the requirements on the plan and does not yet detail much of the plan itself.

WP4-T1-S1-2: Assess D2.3 “Process Definition” with definition of DAS2Vs and V&V steps

WP4-T1-F1-2-1: Very high-level

WP4-T1-F1-2-2: Update expected?

WP4-T1-S1-3: Assess D2.9 “Requirements for Verification & Validation”

WP4-T1-F1-3-1: very very high-level, nothing except requirement numbers for reference relevant for future steps

WP4-T1-S1-4: Assess D2.1 (“Report on Existing Methodologies”)

WP4-T1-F1-4-1: Seems very sketchy

WP4-T1-S1-5: Assess development and V&V activities planned or already on the way for taking them into account in the V&V plan

WP4-T1-S1-5-1: Ask a lot of people (or the right people)

WP4-T1-S1-5-1-1: Design a query email (to be backed up by phone or personal inquiries)

WP4-T1-S2: Organize the writing

WP4-T1-S2-1: Make a detailed work plan

WP4-T1-S2-1-1: Transform the sketch to .tex

WP4-T1-S2-1-2: Revise the structure according to what is expected to be done - accommodating the info on the process (D2.3 -WP4-T1-S1-2) and on ongoing activities (WP4-T1-S1-5).

WP4-T1-S2-1-3: References to the requirements (D2.9 - WP4-T1-S1-3) are to be included

WP4-T1-S2-1-4: Tools and methods

WP4-T1-S2-1-4-1: Format for evaluation (formulate evaluation criteria, D4.1a)

WP4-T1-S2-1-5: Result collection

WP4-T1-S2-1-5-1: Sketch all the formats (purpose)

WP4-T1-S2-1-5-2: Sketch the process of information collection (T4.2 and T4.3 will have to do that)

WP4-T1-S2-1-6: Include section on V&V plan revision

WP4-T1-S2-2: Find contributors

WP4-T1-S2-3: Distribute the work

WP4-T1-S3: Do the work

2 Verification & Validation of the Formal Model

%%To Be Confirmed%%

To ensure the correctness and consistency of the model and its implementation, the validation and verification has to be performed alongside with the modelling process. Thus these tasks will be performed repeatedly during WP3 and will provide feedback to it.

This task handles the verification and validation of the formal model. This will be accomplished by applying the methods chosen in WP4 Task 1 onto the formal model from WP3 using the tool chain developed in WP3. Depending on the chosen approach and applicable tools a variety of verification methods can be applied like:

1. proof technique
2. model checking technique
3. Simulation

As the verification and validation is part of the development chain, this task is being applied iteratively in parallel to the development of the formal model in WP3. The feedback given should focus on the consistency and correctness of the model and development process in WP3. The results of this task are the verification and validation specifications (how to perform the V&V on the formal model), the basic materials (the actual tests cases, checklists, etc.) and the V&V report on the formal model.

The main objective of this task is to guarantee the correctness and consistency of the semi-formal model (SysML) and the fully formal model (SCADE), by applying verification (concerned with building the model right) and validation (concerned with building the right model) techniques. To ensure this, the requirements described in D2.9 “Requirements for Verification & Validation” for the Subset-026 ought to be met by the model. Also, following the standard Technical Specification for Interoperability is mandatory.

As detailed in D2.9, a list of safety requirements will be collected and refined to create requirements and properties. Once these requirements are ready, there are a variety of techniques (described below) that can be used to verify and validate the formal model.

2.1 Proof Technique

A proof is a demonstration that if some fundamental statements (axioms) are assumed to be true, then some mathematical statement is necessarily true. As mentioned in the requirements document produced by WP2, as much as possible, formal proof would then be used to prove that the OpenETCS model never enter a Feared State, as long as the other subsystem (RBC, communication layer. . .) fulfill their own safety properties (axiom describing the environment). Such theorem proving helps to increase our confidence on the specified model. The proof techniques should be integrated in the selected tool chain.

In order to use formal proof to verify if the SFM (Semi-formal model) and FFM (fully formal model) comply with the safety and function requirements (cf. R-WP2/D2.6-02-058), the properties to be proven have to be identified and described. There will be a set of axioms that will describe both functional and/or safety properties of the system. The choice of axioms describing functional and/or safety properties will be provided by safety analysis in an independent way from approaches used to specify, design, validate or verify. It must be noted that the model obtained from the Subsystem Requirements Specification should be verified in this manner at a first stage.

2.2 Model Checking

Model checking is an automatic technique for verifying finite-state reactive systems. As such, one could automatically check if the model specifies most of the requirements of the system, such as the important safety properties described in Task 4.4.

Similar to proof techniques, in order to use model checking to verify if the SFM (Semi-formal model) and FFM (fully formal model) comply with the safety and function requirements (cf. R-WP2/D2.6-02-058), the properties to be proven have to be identified and described. To implement the use model checking, it is mandatory to specify the model using finite-state reactive systems, and they should also provide an intuitive way to express the properties to be model checked. The set of critical requirements to be verified need to be clearly identified. The criteria for the model to be considered a representation of the standard is that all properties are checked. The proposed model checking techniques should be supported in the selected tool chain.

2.3 Simulation

As for simulation, the artifacts should provide means to execute the model. The simulator must be automatically generated, so that, when run against test scenarios (inputs/outputs for the model), we may conclude whether the model follows the specification or not. In particular, it is important to define test scenarios for the safety critical properties. Since, the development within openETCS has to the goal to reach the CENELEC EN 50128 SIL 4 standard, it is highly recommended (cf. SIL 4) that the simulation needs to cover all states, transitions, data-flow, and paths in the model. It would also be desirable to include graphical representation of the simulation/model and also provide a report of the visited components as specified by CENELEC EN 50128 SIL 4.

CENELEC EN 50128 SIL4 also advocates to perform tracing. Being able to trace the requirements that are met during a simulation is also advisable to allow simple requirement coverage.

2.4 Testing Methods

Testing methods will be applied in order to check the correctness of the model with respect to the informal specification and safety requirements. Testing techniques will also be used to check properties that cannot be checked using the proof, model checking or simulation. These techniques will be complementary of the above mentioned techniques. In particular, if a safety requirement/property cannot be proven, testing covering all reasonable possible events/transitions must be used (cf. safety requirements R-WP2/D2.6-02-058.04 and R-WP2/D2.6-02-058.02).

2.5 Other Methods

Reviews, Inspections, static analysis and walkthroughs, mostly manual techniques, are also to be considered for the verification of models.

The inputs required by this task are

- Requirements for the model provided by ERA, SSRS and safety analisys;
- D2.9 - Set of requirements for V&V;
- D7.2 - Report on all aspects of secondary tooling;
- D7.4 – Tool chain first release;
- Model of the system (WP3).

As the goal of this task is the verification and validation of the formal model using the selected tool chain. We will also provide a sequence in which these tools should be executed to check the correctness of the model (e.g., simulation can be used to check properties that cannot be checked by model checking and/or proof techniques).

We propose to start the procedure of tools validation by the application of the following process:

1. To check tools based on proof techniques in order to evaluate what requirements –properties of the model they are able to validate. This evaluation will be based on the work performed in WP7, Task 2;
2. To check what are the properties that can be evaluated using model checking, in particular we will verify what properties can checked by model checking and not by proof techniques;
3. To check what are the properties that can be evaluated using simulation, in particular we will verify what properties can checked by simulation and not by model checking nor by proof techniques;
4. For all tools: to precise if the tools satisfy CENELEC requirements;
5. We will provide a report on the applicability of the tools to check model correctness.

We expect from the partners participating to T4.2, to send to us: - a short description of the methods used for the model description; - first description of the models elaborated by the partners.

Table 2. T4.2 Inputs, Outputs and Deliverables

T4.2 Verification & Validation of the Formal Model				
Type	Description	Due Date	Due Month	status
D	*D 4.4* Final report on Verification & Validation of the model	Jun-2015	T0+36	not started

3 Verification and validation of the implementation and code

The objective of this task is to verify and validate the actual implementation of the formal model. Therefore the tool chain from WP3/WP7 will be used to apply the chosen methods from WP4 Task 1 onto the implementation of the formal model from WP3. The chosen combination of methods and tools in WP4 Task 1 can result in a wide variety of techniques to be used:

- Software-in-the-Loop
- Model-in-the-Loop
- Model-based testing
- Static analysis (e.g. coding guidelines)
- Formal methods (abstract interpretation, deductive verification, model checking)
- Monitoring

Analogue to WP4 Task 2 the verification and validation of the formal model implementation is part of the development chain. Therefore this task runs parallel to the development of the formal model in WP3, and is being applied iteratively. Therefore feedback regarding the validity and correctness is reported to the development process in WP3. The results of this task are the verification and validation specifications, that is,

- how to perform the V&V on the formal model implementation,
- the basic materials (the actual tests cases, checklists, etc.) and

- the V&V report on the implementation of the formal model.

As first steps the relevant properties and techniques concerning the code and implementation are to be identified. We list therefore the following software properties that we think are most relevant for verification & validation:

- functionality
- robustness (absence of runtime errors)
- performance
- real time behaviour
- dataflow
- absence of deadlocks

Table 3 shows the main deliverables of Task 4.3.

T4.3 V&V of the implementation code				
Type	Description	Due Date	Due Month	Status
D	*D 4.2* Interim report on the applicability of the V&V approach	Dec-2013	T0+18	started
D	*D 4.4* Final Report on the V&V of the Implementation	Jan-2015	T0+31	not started

Table 3. Inputs, Outputs and Deliverables of Task 4.3

4 Verification of the Tools and Processes

Since one of the openETCS project goals is to define processes and corresponding tools which are suited to develop ETCS train Onboard Unit software based on a model of the system requirement specifications, the development processes have to follow the CENELEC Standards - especially EN 50128. As this should be done using open source principles and agile development methods it has to be demonstrated that all processes have been applied properly respecting the required principles. Therefore all life cycle stages (SW Requirement Specification, SW Design, SW Coding, etc.) and their related verification and validation activities have to be designed, performed and documented consistently and as required by the CENELEC standards.

The work of this task 4.4 will verify proper design, performance and documentation of the overall development process and its corresponding tools by inspection of the respective documents. According to EN 50129 a safety plan has to be submitted to the relevant safety authorities as documentary evidence that during the development of a safety-related electronic railway system the conditions for safety acceptance have been satisfied. This safety plan consists of a safety case covering evidence of quality management, safety management, and functional as well as technical safety in a structured justification. Since the main product of the openETCS project shall be a non-vital demonstrator implementation it is unnecessary and infeasible for task 4.4 to write a complete safety case. First and foremost the documentation of the actual openETCS development will cover quality management aspects (among others):

1. Documentation of the development process

2. Roles, responsibilities and competences of the involved bodies
3. Traceability during the development
4. Documentation control and Configuration management
5. Fault management
6. Complain handling
7. Modification and change control
8. Tool functionality and tool handling documentation

The safety related verification of tools and processes has to deal with the two aspects, safety management and the functional (and technical) safety. However, the openETCS project will not deliver a vital software implementation, consequently most of the activities will not or only for demonstration purposes be performed in the project. Respectively instead of verifying and documenting the actual activities in a safety case, the work of task 4.4 will rather be to present a generic justification structure for a safety case, which can be used in following projects or by the industrial partners for the development of an openETCS based Onboard Unit. The partial execution of safety activities shall be used to demonstrated the structure and to show their serviceability.

To reach this goal the safety management team work in task 4.4 will design a safety plan to outline the overall structure of the safety management and the pursued way to demonstrate the functional safety. Accordingly the sequence of safety activities will be identified including all verification and validation activities for safety requirements. Overall the safety plan will describe the following points:

1. Safety specific roles, responsibilities and competences of the involved bodies
2. Principles and overall structured process of the safety management
3. Requirements on the different tools supporting the development process and criteria for the tool categorization
4. Safety related activities and all corresponding artifacts (required inputs/outputs, legislative documents and further needs) to prove functional (and technical) safety (including for the supporting tools)
5. Structure of the safety case based on the documentation of the safety activities and their supporting tools
6. Principles and procedures for preparing the safety case
7. Specific procedures for maintaining the functional safety and the corresponding safety documents over time

All these activities are closely connected to the design, verification and validation activities and therefore have to be defined considering their inputs. Respectively the task 4.4 has to work in close contact with all other WPs and all especially with all tasks in WP 4. Concerning the applied tools, it is likely that 3rd parties have to be engaged to perform the verification and validation or deliver the needed documentation like technical specification, potential hazards and their risk in respect to the overall development process.

Following the safety plan sample activities shall be chosen to be performed on part of the Onboard unit. This shall give evidence that the presented safety management is exercisable and can be performed by the used tools. Additionally it shall demonstrate the safeguards used for individual safety properties to ensure functional and technical safety. Since in a model based development the hazard identification, risk assessment and safety requirement verification is very closely related to the design activities and the supporting tools, safety activities have to be performed mainly by other teams in the development process. The safety team of task 4.4 itself can only guide these activities and process the results to present them in the safety case.

The following table **??**gives an overview concerning the Inputs and Outputs of task 4.4.

Table 4. T4.4 Inputs, Outputs and Deliverables

T4.4 Verification of the Tools and Processes				
Type	Description	Due Date	Due Month	status
→	*D 1.3.1* Project Guide on Quality Assurance	Jun-2013	T0+12	no
→	*D 2.4* Methods definition	Feb-2013	?	no
→	*D 2.6-9* Set of Requirements	Jun-2013	?	no
→	*D 7.1* Report on the final choice(s) for the primary tool chain (means of description, tool and platform)	Jun-2013	?	no
→	*D 7.2* Report on all aspects of secondary tooling	Jun-2013	?	no
→	*I: O 7.2.8* Safety analyses tools choices	Jun-2013	?	no
→	*D 7.3.1.2* Tools Interoperability Description	Jun-2013	?	no
→	*I: O 7.3.1* Tool chain development plan (or equivalent)	Jun-2013	?	no
→	*I: O 7.3.2* Specification of tool interoperability mechanisms	Jun-2013	?	no
→	*I: O 7.3.4* Specification of primary and support tool chain architecture and its embedding into the platform	Jun-2013	?	no
→	*D 7.3* Tool Chain Qualification Process Description	Jun-2013	?	no
→	*D 7.4* Tool chain first release	Feb-2014	T0+20	no
→	*WP3* Feedback concerning the quality and safety management processes	Jun-2013	?	no
→	*WP3* Feedback concerning potential hazards	Jun-2013	?	no
→	*D 4.1* Report on V&V Plan & Methodology	Jul-2013	T0+13	no
→	%%I: O Task 4.2%%	Jul-2013	?	no
→	%%I: O Task 4.3%%	Jul-2013	?	no
→	*WP4 - T 4.5* Feedback concerning documentation quality	Jul-2013	?	no
←	*O 4.4.1* Safety Plan	Oct-2013	T0+16	started
←	*O 4.4.2* Report on safety demonstration activities	Feb-2014	T0+20	not started
D	*D 4.4* Final report concerning the Safety Case	Jun-2015	T0+36	not started

5 Internal Assessment

One of the major point for a SIL4 compliant Software is the Whole Software Development Project Assessment by a Safety Authority (e.g CERTIFER in France, TÜV in Germany). As none of these companies are involved in openETCS Software Development assessment, the Internal Assessment objective is to simulate a real Assessor's tasks during the whole Open ETCS Software Development activities.

An assessment is a ¨ Process of analysis to determine whether software, which may include process, documentation, system, subsystem hardware and/or software components, meets the specified requirements, and to form a judgment whether the software is fit for its intended purpose. Safety assessment is focused on but not limited to the safety properties of a system.¨

5.1 Assessment tasks

The Assessor shall write a Software Assessment Plan. It is like an assessment process which is linked to the software development process. More precisely, he shall explain the tasks needed to assess the software of the project OpenETCS.

Note: The Verifier shall write a Software Assessment Verification Report, as required in the standard EN50128, to verify in the first time that the Software Assessment Plan meets the general requirements for readability and traceability.

During the software development, he shall evaluate the software verification and validation activities. We propose that the Assessor intervenes at least seven times during the software development process (this is equivalent to one time at least by Work Product).

Note: the numbers of work packages are not given in the chronological order, e.g. WP1 is performed during all the development process and WP5 occurs before the end of WP4.

During WP1: Project Management.

The Assessor is able to assess:

- The Quality Assurance
- The capability of the Project Manager and the quality of his deliverables

The Assessor shall assess the Software Quality Plan. We propose that he gives a formal approval of this document.

During WP2: Requirements for Open Proof.

The Assessor is able to assess:

- The System requirements specification, including:
  - functions and interfaces;
  - application conditions;
  - configuration or architecture of the system;
  - hazards to be controlled;
  - safety integrity requirements;
  - apportionment of requirements and allocation of SIL to software and hardware;
  - timing constraints
- The software requirements specification,
- The software architecture and design specification,
- The software component specification,
- The personnel key roles, responsibilities and competence,
- The Quality Assurance

Nevertheless, he shall assess the implementation of both activities and deliverables of WP 2.

During WP3: Modeling of (part of) ETCS specification.

The Assessor shall evaluate the software implementation respectively the software modeling. Furthermore, he is able to assess:

- A part of the life cycle and the documentation,
- The Quality Assurance,
- The personnel roles and responsibilities and competence.

The Assessor shall assess the implementation of both activities and deliverables of WP 3.

During WP4: Validation & Verification Strategy.

The Assessor shall assess:

- the Software Verification Plan and the Software Validation Plan,
- the Quality Assurance.

We propose that he gives a formal approval on these documents. He shall mainly evaluate the verification activities and the implementation of both activities and deliverables of the WP 4.

During WP5: Demonstrator.

The Assessor shall assess the specific openETCS software. Indeed, before the beginning of the validation activity (WP4), the Assessor shall assess the Software Integration Test Report to give or not the approval for software validation. This point is the Validation first step (the previous steps are related to the verification).

During WP6: Dissemination, Exploitation and Standardization.

The Assessor shall verify that the software maintenance plan is written and compliant with the software safety integrity level (SIL4).

During WP7: Language, Tool Chain and Open source Ecosystem.

The Assessor shall assess the developed tool chain according to Tool class T3 of EN 50128:2011. The other tools (T2) have to be assessed as well, but the effort is liter regarding the T3 assessment effort.

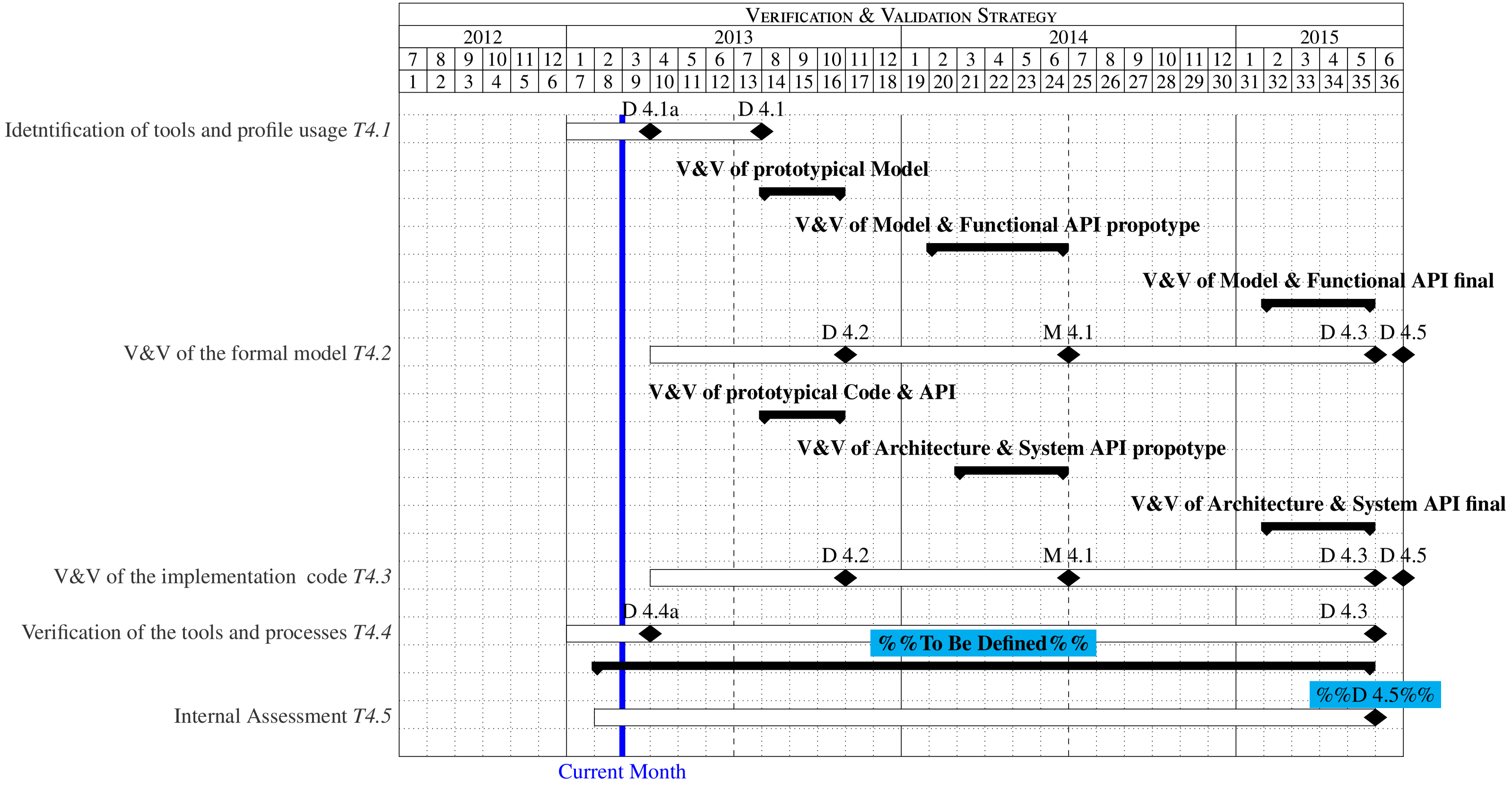
At the end of the software development process, the Assessor shall perform a final assessment. Indeed, he shall evaluate that the life cycle processes and products resulting are such that the software is of the defined software safety integrity level and fits for its intended application. All the steps of assessment performed during the software development process shall be gathered in the Software Assessment Report. This report could be updated all along the process.

Note: the Software Assessment Verification Report permit to verify the internal consistency of the Software Assessment Report.

Table 5. T4.5 Inputs, Outputs and Deliverables

T4.5 Internal Assessment				
Type	Description	Due Date	Due Month	status
D	*D 4.5* Quality recommendation to prepare the Assessment	29/03/2013	March	

6 GANTT chart


VERIFICATION & VALIDATION STRATEGY
Idetntification of tools and profile usage T4.1
D 4.1a
D 4.1
V&V of prototypical Model
V&V of Model & Functional API propotype
V&V of Model & Functional API final
V&V of the formal model T4.2
D 4.2
M 4.1
D 4.3
D 4.5
V&V of prototypical Code & API
V&V of Architecture & System API propotype
V&V of Architecture & System API final
V&V of the implementation code T4.3
D 4.2
M 4.1
D 4.3
D 4.5
Verification of the tools and processes T4.4
D 4.4a
D 4.3
%%To Be Defined%%
%%D 4.5%%
Internal Assessment T4.5
Current Month