ITEA2 Project
Call 6 11025
2012 – 2015

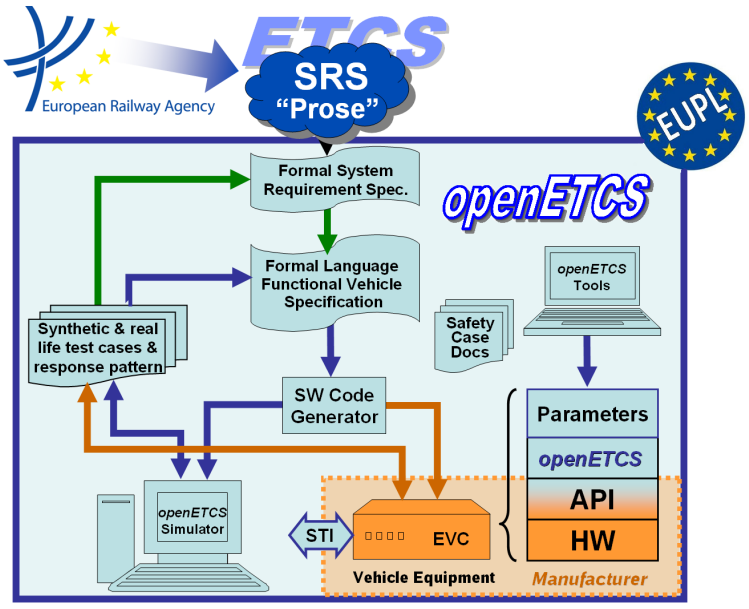
Work Package 1: “Governance”

openETCS Software Release and Deployment Plan

Version 0.1.0

Bernd Hekele (DB-Netz AG) 06. May 2014

Contributions by:



Funded by:

Federal Ministry of Education and Research
Région de Bruxelles-Capitale
GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

This page is intentionally left blank

Work Package 1: “Governance” **OETCS/WP3/D3.6.1.1**
06. May 2014

openETCS Software Release and Deployment Plan

Version 0.1.0

Bernd Hekele (DB-Netz AG)
Contributions by:

DB-Netz AG
Völckerstrasse 5
80939 München, Germany
eMail:bernd.hekele@deutschebahn.com

Part of Quality Assurance Plan

Prepared for openETCS@ITEA2 Project

Abstract: This document describes the strategy and plan of software releases in the openETCS ITEA project.

Table of Contents

Figures and Tables

Figures

Tables

Document Control

Document information	
Work Package	WP1
Deliverable ID or doc. ref.	D1.3.1
Document title	openETCS Software Release and Deployment Plan
Document version	0.1.0
Document authors (org.)	Bernd Hekele (DB-Netz AG)

Review information	
Last version reviewed	–
Main reviewers	–

Approbation			
	Name	Role	Date
Written by	Bernd Hekele	WP3 Task 3.6 Leader	May 2014
Approved by			

Document evolution			
Version	Date	Author(s)	Comment
00.01	17.04.2014	Bernd Hekele	Document creation
00.01	06.05.2014	Bernd Hekele	First comments on Structure included

1 Introduction

From its nature openETCS is not a simple development project. In openETCS other aspects have a similar weight compared to the actual developed product:

- Clarification of the SRS, Clearing of Findings.
- Introduction of a formal model to be used as a quasi standard extension for the SRS
- Introduction of Agile Methods in Development.
- Introduction of openSource and openProofs as a method to produce safety critical software.
- Production of a software build to show the standard alive in a ETCS demonstrator.

These additional tasks are part of the research character of the openETCS project.
These research aspects have an implication on this release plan as well:

1. We do not plan to deliver a safe product to run a train.
2. We do not start with fixed processes at the beginning of the project. Processes are developed hand-in-hand with the progress of the development work.
3. We have some additional outputs as a result of the release which are not common to this class of development.

This document (the plan) also will be developed further with each iteration in the project. An updated and enriched version is planned:

1. At the end of the first iteration
   This update will define deliverables of regular iteration releases in detail:
   - When do we release?
   - Where do we release?
   - What do we release?
   - In which quality do we release?
     Mandatory Checks
   - Procedures to report findings of iteration releases.
2. At the end of the each following iteration
   We will incorporate our lessons learned and findings of the Scrum Rehearsal.

1.1 openETCS Release Stakeholders

Unlike other development projects the modelling result (i.e., the content of the releases) is not directly feeding into a product. But, we have to support our internal stakeholders for the modelling results:

- Verification & Validation (WP4).
- Demonstrator (WP5), i.e., the code goes directly into a train simulator.
- The Railway Community (represented by ERA), who especially is interested in the clarification of the standards.

1.2 Glossary

API: Application Programming Interface. In the project, the API defines the interface of the EVC software to the operating system and hardware. *The exact nature of the API still needs to be defined, whether it should be seen as a specification or as an implementation has yet to be resolved.*

Build: A compiled package of software. If possible, the package will be integrated in a binary ready for testing. Builds can be produced frequently to support the integration of software parts [daily Build]. In openETCS a build has to be produced at least at the end of a sprint.

Deployment:

Release: A release is a build with a qualified status by verification and validation.

Iteration: A number of sprints defined to reach a defined scope of the model. In openETCS we plan to have a release at the end of an iteration.

Sprint: Phase in agile development. See Definition of Scrum in the QA-Plan. In openETCS modelling, the sprint lasts 2 weeks. At the end of a sprint, the artefacts will be produced in a sprint build. To get the stamp "Done" the function of artefacts have to be demonstrated in the sprint review meeting.

SRS: (ERTMS/ETCS-) System Requirements Specification.

SW: Software

2 The Release Plan

2.1 The Release Template

For documentation of releases a template is to be used. You can find the template in this location: `https://github.com/openETCS/governance/blob/master/SoftwareReleaseAndDeploymentPla openETCSReleaseTemplate.tex`.

2.2 openETCS Timeline: Development Phases

openETCS modelling introduces agile development methods. Basis for a release plan in an agile world is visible in the following planning concepts (please, refer to figure 1). The figure shows the tools in openETCS and artefacts needed as input our output from the tools. In ´general, all results from the production process are provided as a part of releases, e.g., generated code, compilation listings, object code etc.

- **Sprint**
  An openETCS modelling sprint is planned to last for 2 weeks. During the sprint a team works on priorized items taken from the modelling backlog. Each item has a "Done" criteria to be reached at the end of the sprint. The "done" criteria is the target for validation which is done as a part of the sprint. To support agile we need some principles:
- **Daily Builds**
  during the Sprint. The daily build has to integrate all software parts of openETCS modelling. The concept for openETCS has to be elaborateds and will be started as soon as sufficient code has been generated and integrated (second iteration).
- **Automated Tests**
  for validation of the daily build and - more important - at the end of the sprint for the sprint review.
- **A sprint release** for each successfully completed sprint.
- **Iteration**
  The openETCS modelling will increase the coverage of the SRS functionality in iterations. The duration of an itreration is defined by the number of sprints planned for the iteration. The content of the iteration will be defined at the end of the previous iteration and is visible in the srs-analysis issues tracker: `https://github.com/openETCS/srs-analysis/issues/milestones`. Iterations do not have a fixed length, however, we shall plan for about 4 iterations per year. Each iteration has an iteration release as a result. The release might be taken by WP5 for building a new version of the openETCS demonstrator.
- **First Iteration**
  The first Iteration is defined with a strictly limited scope in order to run the full modelling cycle and producing the full set of artefacts in onpenETCS. Target of the iteration is the proof of the concept in a small model.
- **Second Iteration**
  The second iteration shall be defined from an operational point of view. The scenario shall cover - under well defined operational limitations - all functions needed to start the train, control speed and stop the train. Details will be defined in an openETCS modelling workshop based on the recommendation of the openETCS stakeholders (WP4, WP5, operators).

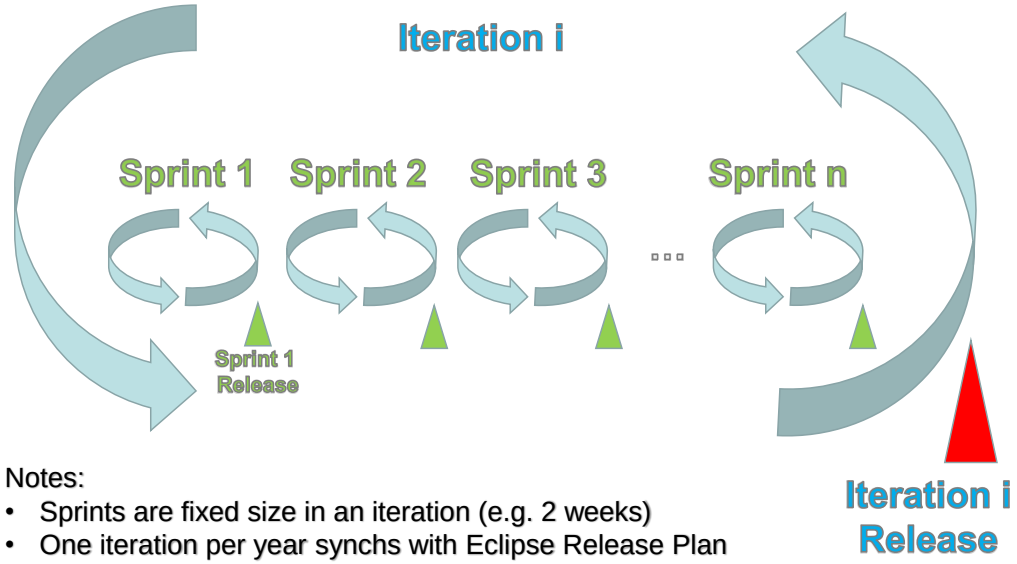


Figure 1. Sprints, Iterations and Releases

- **Release in the context of Eclipse**
  Each year the Eclipse Foundation produce a release on a coordinated schedule:
  `http://wiki.eclipse.org/Simultaneous_Release`
  openETCS will synchronise a yearly release with the Eclipse schedule. The openETCS release covers toolchain as well as modelling results. In modelling, a convenient iteration will be chosen for this approach.

Setting for the first iteration: 6 sprints, each sprint 2 weeks.
Proposed setting for the second iteration: 10 sprints, each sprint 2 weeks.

2.3 openETCS Scope of the Release

In this section, we define the scope and the quality of the openETCS modelling releases according to the use-case. Links to the location in the openETCS repositories will be provided as soon as they are defined. Figure 2 indicates the artefacts and the respective use-case. The current diagram is limited to modelling.

- **Daily Build Report**
  The Build Report, i.e., details of the server are to be clarified. At the server, the following information is available:
  1. Build Highlights (Entry Page on wiki with summary information)
  2. Information on which version of model parts have been used to make the build.
  3. Generated C-Code
  4. Generation Result (Scade Suite Output)
  5. Compilation Result and Object Code

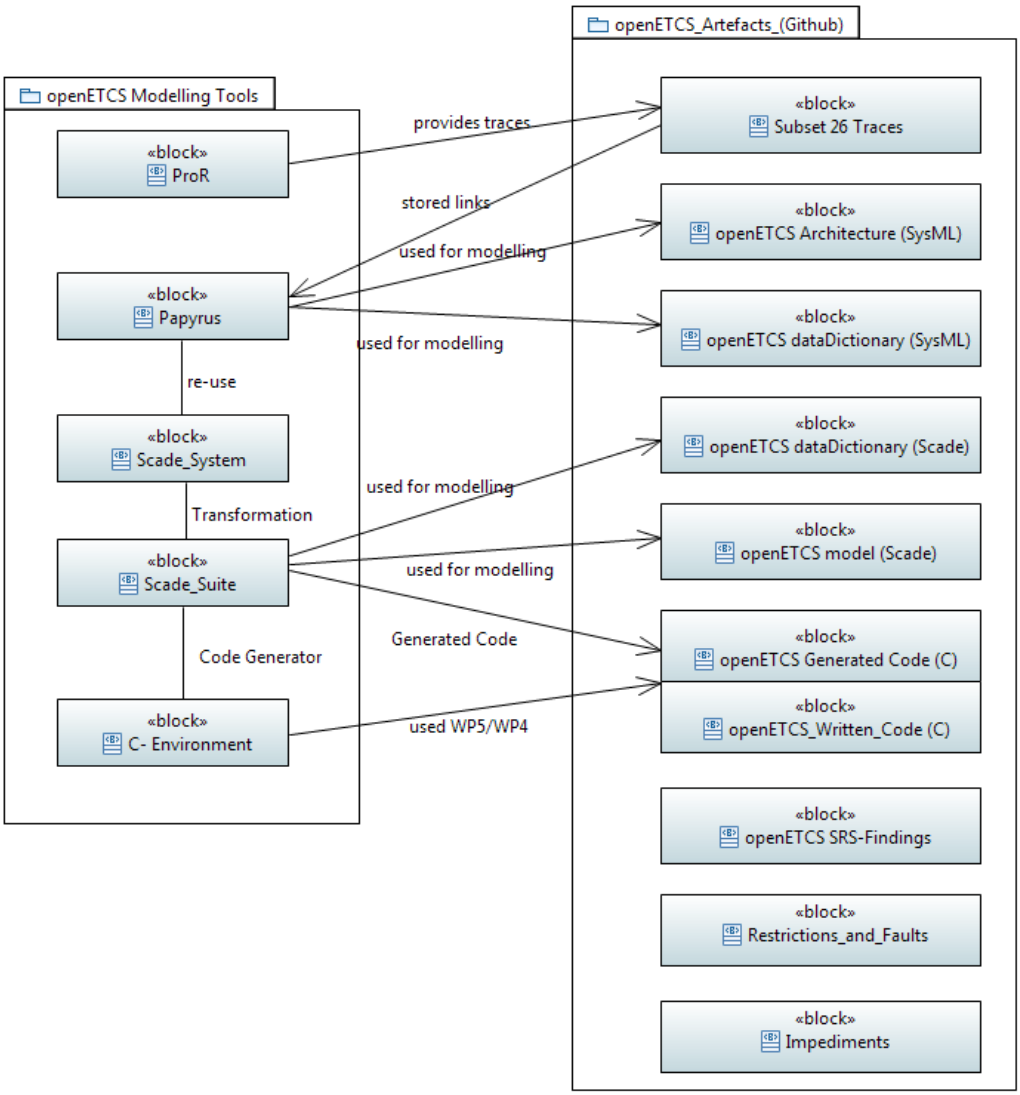


Figure 2. openETCS Tools and Artefacts

6. Test results of automated test-cases

- **Sprint Release Report**
  The sprint release, i.e., details related to the sprint result:
  1. Build Report of final build of the sprint
  2. Updated SysML model, Synchronisation of Scade Suite and Scade System model
  3. Updated Sprint Backlog incl. "Done" Status of items
  4. Restrictions and Fault reports related to failed backlog items
  5. List of Impediments
- **Iteration Release Report**
  1. Sprint Release Report of last Sprint of the Iteration
  2. Sprint Retrospective Result
  3. The SysML and the Scade Suite model
  4. validation results
  5. List of open Errors and Restrictions (with Status and Priority)
  6. Update on SRS Findings
- **Yearly Eclipse Synchronised Release**
  1. Iteration Release Report of relevant iteration
  2. update of project status

3 The Deployment Plan

In this project, deployment covers the delivery of a release to the demonstrator work package. The openETCS model is part of the simulation framework - not of a real train. Since openETCS follows the Eclipse philosophy a yearly release of the openETCS model e.g., synchronised with the Eclipse coordinated releases, is planned.

The plan relevant for deployment will be available at the end of the second iteration. The plan will answer the questions:

- When do we deploy?
- Where do we deploy?
- What do we deploy?
- In which quality do we deploy?
- Procedures to report findings.

4 References

- **D1.3.1**: Quality Assurance Plan
- **D1.3.1**: Software Release Template