OETCS/WP2/D2.3b

openETCS

ITEA2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

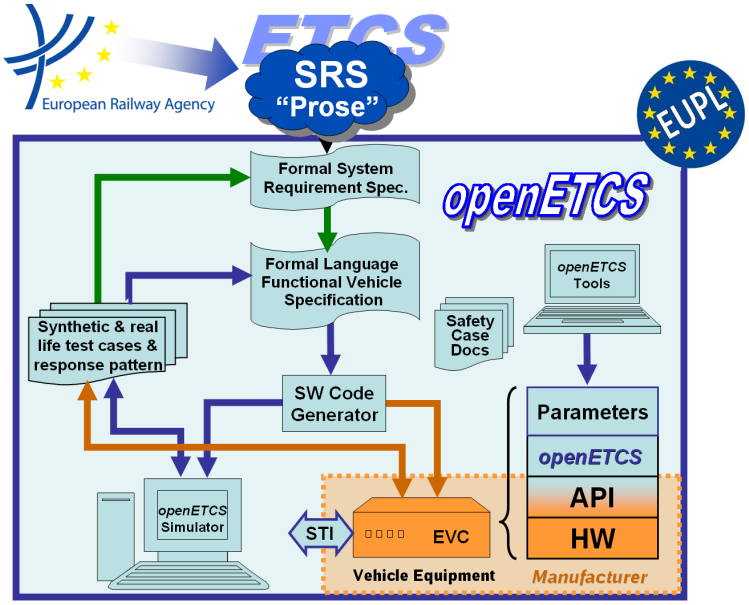
ITEA2 Project
Call 6 11025
2012 – 2015

Work-Package 2

Scrum Process in openETCS Project

Abdelnasir Mohamed and Jan Welte

August 2015
Revised December 2015



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 2

OETCS/WP2/D2.3b
August 2015
Revised December 2015

Scrum Process in openETCS Project

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature Abdelnasir Mohamed (AEbt)	signature () ()	signature () ()	signature Klaus-Rüdiger Hase (DB Netz)

Abdelnasir Mohamed

AEbt Angewandte Eisenbahntechnik GmbH
Adam-Klein-Str. 26
90429 Nürnberg, Germany
eMail: abdelnasir.mohamed@aebt.de
WebSite: www.aebt.de

Jan Welte

Technische Universität Braunschweig
Institute for Traffic Safety and Automation Engineering
Hermann-Blenk-Str. 42
38108 Braunschweig, Germany
eMail: openetcs@iva.ing.tu-bs.de
WebSite: www.iva.ing.tu-bs.de

Output Document

Prepared for openETCS@ITEA2 Project

Abstract:

Table of Contents

Figures and Tables

Figures

Tables

Document Control

Document information	
Work Package Deliverable ID	
Document title Document version Document authors (org.)	openETCS 1.0 Abdelnasir Mohamed (AEbt) and Jan Welte (TU-BS)

Review information	
Last version reviewed	0.1
Main reviewers (org.)	

Approbation			
	Name	Role	Date
Written by	Jan Welte	WP4-T4.4 Task Leader	March 2015
Approved by	–	–	

Document evolution			
Version	Date	Author(s)	Justification
0.1	18/08/2015	Jan Welte	Document creation
0.2	19/10/2015	Abdelnasir Mohamed	Chapter 1.1
0.3	28/01/2014	Jan Welte	Editing of Chapter 1.4, 3 and 4
0.4	28/11/2015	Abdelnasir Mohamed	Editing of Chapter 1.2, 1.3, 1.4.2, 1.4.3 and 3
1.0	08/12/2015	Jan Welte	Minor corrections in Chapter 1.1, 1.4, s3 and 4

1 Introduction

1.1 Purpose

The purpose of this document is to define and describe the agile work-flow of the software development process in the openETCS project. As the outcomes of the project, the openETCS software and the openETCS development tool chain, are precisely defined, the process of how these two artifacts are being produced in detail has not been clear at the beginning of the development process. In the conventional software development process, e.g. with a V-Model, the life cycle of the development is implemented in closed non re-definable phases, which shall be processed consecutively.

Due to the complex nature of the openETCS project as a Research and Development (R&D) Project an agile methodology is needed, which in it's framework facilitates continuous refinements and improvements throughout the whole openETCS software development process. In this agile way of development complex tasks are broken down into smaller tasks (increments), which shall be executed in short time frames called Sprints. This document describes the agile framework in the openETCS development process applying SCRUM.

1.2 Reference Documents

This document essentially refers to the following standards, ETCS specification documents and openETCS project documents.

- **ISO 9000** — 12/2005 — *Quality management*
- **ISO 9001** — 12/2008 — *Quality management systems — Requirements*
- **ISO 25010** — 03/2011 — *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*
- **CENELEC EN 50126-1** — 01/2000 — *Railways applications — The specification and demonstration of Reliability, Availability, Maintenability and Safety (RAMS) — Part 1: Basic requirements and generic process*
- **CENELEC EN 50128** — 10/2011 — *Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems*
- **CENELEC EN 50129** — 05/2003 — *Railway applications — Communication, signalling and processing systems — Safety related electronic systems for signalling*
- **CCS TSI** — *CCS TSI for HS and CR transeuropean rail has been adopted by a Commission Decision 2012/88/EU on the 25th January 2012*
- **SUBSET-026** 3.3.0 — *System Requirement Specification*
- **SUBSET-091** 3.2.0 — *Safety Requirements for the Technical Interoperability of ETCS in Levels 1 & 2*
- **SUBSET-088** 2.3.0 — *ETCS Application Levels 1 & 2 - Safety Analysis*

- **OpenETCS FPP** — *Project Outline Full Project Proposal Annex OpenETCS* – v2.2
- **OpenETCS D1.3.1** – Project Quality Assurance Plan
- **OpenETCS D2.2** – Report on CENELEC standard
- **OpenETCS D2.3** – Definition of the overall process for the formal description of ETCS and the rail system
- **OpenETCS D2.4** – Definition of the methods used to perform the formal description
- **GitHub: WP3 Wiki** – WP3 Scrum Process
- **GitHub: WP4 Wiki** – Scrum of Scrums
- **The Scrum Guide** – The Definitive Guide to Scrum: The Rules of the Game, July 2013

1.3 Glossary

ETCS	European Train Control System
ERA	European Railway Agency
EVC	European Vital Computer
FMEA	Failure Mode Effect Analysis
OBU	On-Board Unit
PO	Product Owner
PB	Product Backlog
PBI	Product Backlog Item
QA	Quality Assurance
SIL	Safety Integrity Level
ScM	SCRUM Master
SoS	Scrum of Scrums
SRS	System Requirement Specification
V&V	Verification & Validation

1.4 Background Information

1.4.1 Agile Development

At the beginning of the 1990s software developers started to see the traditional software development methods as inadequate and too inflexible to deal with the increasing dynamic environment. The development methods derived under this concept are today referred to as agile methods. The word agile is in this context understood in the sense of movable or nimble. The basic concept of agile methods is software development process, which is able to adapt to changing requirements from needs and wishes of the client while - or perhaps on these grounds - having a stable development.

The traditional methods are based on the premise, that once a system has been specified and designed it can be implemented without major modifications. In accordance the requirements are collected as first as detailed as possible in traditional methods. Afterwards, a number of

predefined artifacts is created in consecutive phases to increase the granularity up to the source code. In an agile understanding the empirical development process should be defined only as strict and distinct as it appears necessary and appropriate to ensure the project's success. By doing so agile software development implicitly address quality assurance (QA) through continuous feedback and recurrent tests, which are carried out during the development. In contrast, the QA is explicitly achieved through process standardization and testing procedures in the traditional software development.

The various agile methods or their process models present themselves thereby not uniform as they differ considerably in their conceptions. Each agile method applies in their own way steps to allow learning and to adopted changes. Established agile methods are Adaptive Software Development, Extreme Programming or SCRUM. The openETCS project bases it's agile development on the SCRUM concept, which was devised in the 1990s by Ken Schwaber and Jeff Sutherland and uses ideas from the lean production concept. The following subsection introduces the fundamental principals and roles of SCRUM.

1.4.2 SCRUM Definition And Roles

This section should be considered as an introduction for the following chapters to fit the content of this document. For more detailed information about SCRUM the *Scrum Guide* is advisable to read. The SCRUM Guide defines SCRUM as "*A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value*". It is "*simple to understand*" but "*difficult to master*". For this framework three roles are defined: the Product Owner (PO); the SCRUM Master (ScM) and the Development Team. All three roles together make the SCRUM Team.

1.4.3 General SCRUM Framework

Scrum process starts with a Product Backlog (PB), this Product Backlog contains an ordered list of requirements that is maintained for a product. The Product Backlog Items (PBIs) are ordered by the Product Owner based on considerations like risk, business value, date needed, etc. The Product Owner is the responsible of the product backlog and the prioritization of PBIs.

After the defining of the Product Backlog the next task is to create the Sprint planning meeting. This meeting is organized at the beginning of the Sprint cycle. The objective of this meeting is to define the PBIs to be done in the following Sprint. The Sprint is the time period in which development occurs on a set of backlog items that the team has committed to (also commonly referred to as iteration).

The result of PBIs selected for implementation in one Sprint is called the Sprint Backlog. The Sprint Backlog is the list of work the Development Team must address during the next Sprint. This list is derived by selecting product backlog items from the top of the product backlog until the Development Team fills it's capacity and make sure it has enough work to during the next Sprint.

Each day during the Sprint, a project team communication meeting occurs. This is called a daily Scrum Meeting and has specific guidelines. The Scrum Meeting is organized by the Scrum Master and the participants respond to three questions:

1. What have you done since yesterday?

2. What are you planning to do today?
3. And any impediments/stumbling blocks?

At the end of the Sprint, the result of the PBIs implemented is called the increment (potentially shippable increment), this is the sum of all the Product Backlog items completed during a Sprint and all previous Sprints. The increment must be in a usable condition regardless of whether the Product Owner decides to actually release it.

At the end of a Sprint cycle or iteration, two meetings are held; the Sprint review meeting and Sprint retrospective. The approximate duration of the spring review meeting will be no more than 4 hours and it will be moderated by the ScM. In this meeting two main activities will take place. One is to review the planned work that has been completed and the work that wasn't. And the second activity is to present to the Product owner the work done. The other meeting after an iteration is the spring retrospective. This meeting will be moderated by the ScM and it's main objective is the continuous process improvements. Two main questions are asked to all participants in this meeting:

- What went well during the Sprint?
- What could be improved in the next Sprint?

2 Agile Development in openETCS

The openETCS Project is concerned with the development for the kernel software of an ETCS on-board unit (OBU). The European Vital Computer (EVC) as the central part ot the OBU contains the core control functionality for the train movement and therefore has a SIL 4 classification according to the applicable CENELEC standards. Respectively, its software development has to satisfy the SIL 4 process requirements of the EN 50128:2011 standard.

Although the openETCS project itself only aims to provide a non vital functional reference implementations, which itself does not require SIL 4 methods, the project has the goal to provide a development methodology applicable for SIL 4 development. This process makes use of model-based development and agile development principals, while following the QA principals which are emphasized by the EN 50128.

As the CENELEC standards have been defined based on a traditional V-Model development process, a matching of roles and the lifecycle is needed to integrate agile SCRUM principles in the openETCS development process. This mainly involves addressing tasks of several phases at the same time and thereby creating artifacts in iterative steps. In addition the required organizational structures for SIL 4 development mainly ensuring independence between design, testing and V&V have to be established in the agile work organization.

2.1 Roles and Responsibilities

Although the CENELEC standards and the SCRUM development process both define specific roles these have a different focus. The CENELEC roles are specified to establish responsibilities which are going along with specific tasks. The clear separation of certain roles between different personnel which can also be under different organizational structures shall ensure independence as the basis for QA. SCRUM roles are purely defined from a process view. They shall ensure that the costumer needs are always the focus point of all activities and that the agile principals of learning and adopting are always applied in a consistent way. Consequently, these roles for the most part are not separating working tasks performed during the development.

Therefore, the QA aspect of the CENELEC roles can be fulfilled in a SCRUM working process by ensuring the required personnel and organizational independence during the SCRUM work allocation. As the SCRUM process is focused on small communicating teams, it is reasonable to further strengthen the personnel independence by addressing larger work portions like validation in separate SCRUM teams. By having several SCRUM Teams working in parallel it is necessary to establish means, which continuously allocate and groom the task between those teams and incorporate the lessons learned. The openETCS project uses a so called SCRUM of SCRUM (SoS) in which a representative of each SCRUM Team and the overall product owner pre-groom task and allocate them between the SCRUM Teams, e.g. modeling and V&V. Afterwards each SCRUM team grooms all tasks further and addresses them in their Sprint planning. Feedback from the Sprint review is channeled back to the SoS. Only in the SCRUM Team specific people are assigned to as tasked in accordance with their qualification, which can provide the same separation and qualification assurance as required in a SIL 4 development.

2.2 Lifecycle

The SCRUM process corresponding to the agile development principals does not explicitly defines artifacts which have to be delivered during the software lifecycle. The SCRUM process only provides a process framework in which all needed artifacts to satisfy costumer needs shall be developed and maintained. In contrast is the CENELEC lifecycle explicitly stating a number of artifacts which shall be provided during the lifecycle. Respectively, these artifacts are provided during the openETCS development as it is presented in D2.3a and figure 1.

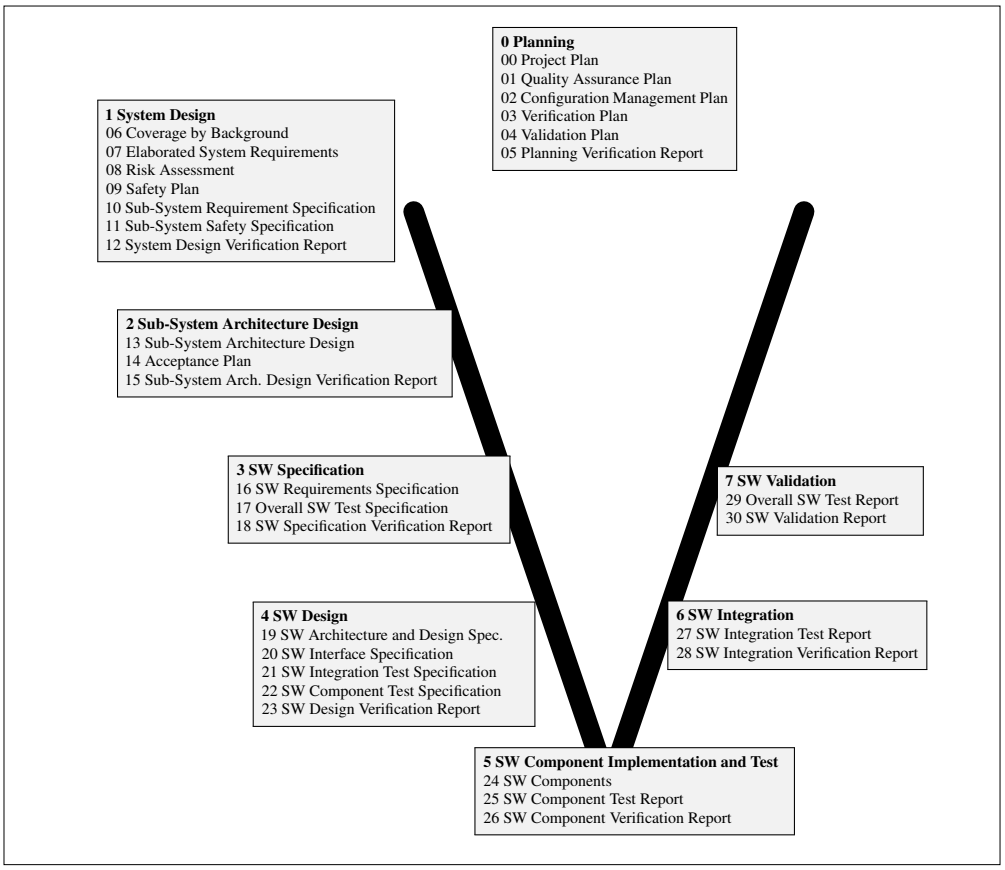


Figure 1. openETCS Development Lifecycle

The agile development is applied as these artifacts are not produced one by one working cycle and without exchanges. In fact, the artifact are more or less a result of tasks essential to reach the final goal of a functional ETCS OBU kernel software. All artifacts defined in D2.3 are either an intermediate product like the SysML Architecture Model or the SCADE Software Specification Model, or they are needed documentation and results for QA measures like plans and reports.

As the agile development is based on continues development, which repeatedly takes feedback and changes into account, the artifacts are completed over time with the ongoing work. Thereby, the system is developed in iterations, which has the consequence that also the corresponding artifacts are produced over time. Thereby, each iteration strongly requires V&V steps for every artifact and simulation to allow costumer feedback. These QA measures are consistent with the CENELEC standards lifecycle principal which require verification and testing reports for every artifact and validation for the overall system. The critical step for the agile development is to ensure consistency between all related artifacts and to establish a capable configuration management to ensure that all findings and the resulted changes are taken into account properly in all following iterations. To allow fast feedback and to be able to repeat these activities with every

iteration an efficient tool support and test automation is needed in agile development. Figure 2 demonstrates the general SCRUM concept in openETCS.

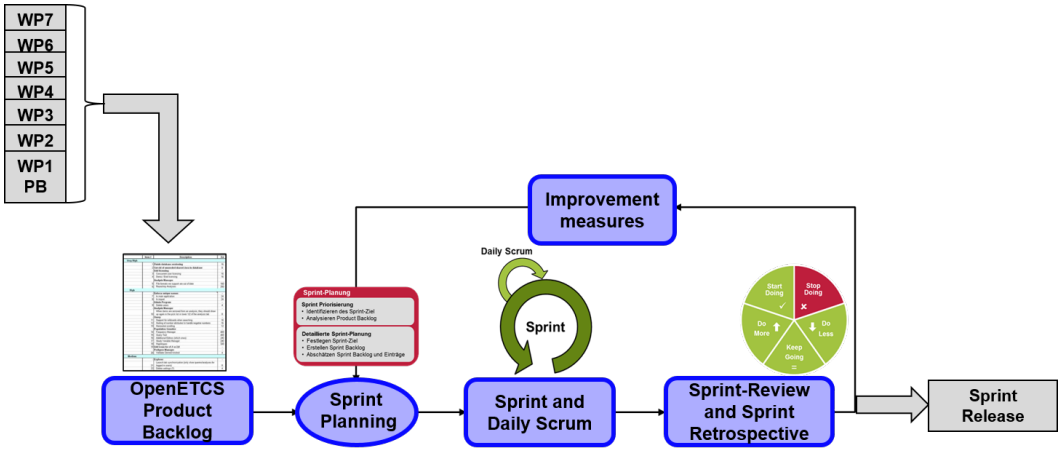


Figure 2. openETCS SCRUM Sprint

Since the V&V activities are extensive for SIL 4 development the openETCS development separated simple test done by the development SCRUM Team itself and the independent V&V checks performed by the V&V SCRUM team. There interaction is coordinated through the SCRUM of SCRUM and releases as shown in section 4.

On the whole the openETCS development process follows the EN 50128 V-model lifecycle for railway software development by using SysML and SCADE models as the main means of transcriptions during the development. Respectively, the EN 50128 artifacts are matched to these model-based design as presented in D2.3a. The agile process of SCRUM is applied to develop these artifacts in incremental steps which changes the content in iterations. The basic relation between the artifacts and corresponding tasks follow the CENELEC QA concept.

3 openETCS SCRUM Process

In openETCS there are three levels of processes that apply SCRUM:

- The first level is the openETCS Project level, where Project Coordinator is the Product Owner. In this level the SCRUM meetings are organized every Friday.
- The second level is the Work Package level, where Work Package Leader is the Product Owner. In this level SCRUM meeting depends on WP, but at least two SCRUM meetings are organized every week.
- The last level is the Task level, where the Task Leader is the Product Owner. In this level SCRUM meeting period is variable and depends on the task.

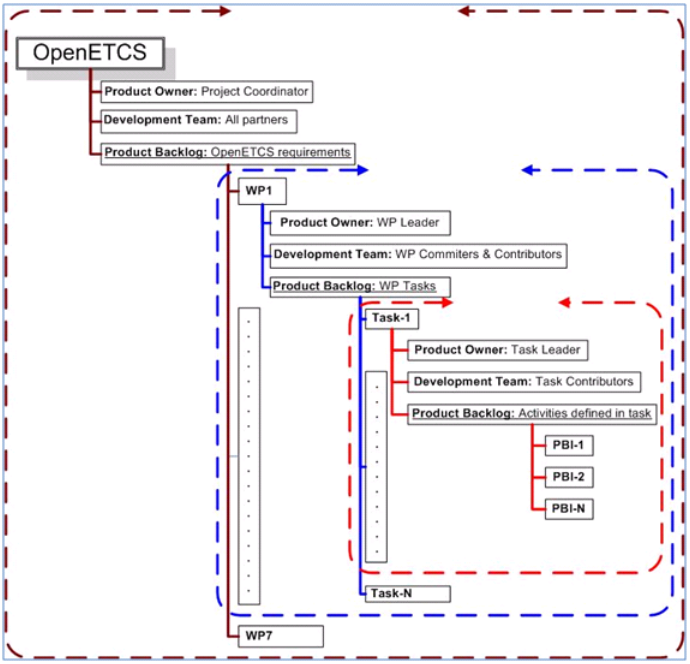


Figure 3. openETCS Project Structure

Compliance to SCRUM Requirements is achieved by means of

- Each Work Package/Top-Project Leader is the SCRUM Product Owner of the corresponding WP/Top-Project results and maintains the corresponding backlog
- Each Project/Task Leader is the SCRUM Product Owner of the corresponding Tasks results and maintains the corresponding backlog
- The Project Coordinator is the SCRUM Product Owner of the project results and maintains the project results backlog.
- Weekly meetings are maintained to find and report on impediments, assess progress, promote cross-collaboration, plan next steps and therefore, maintain the corresponding backlog.

  - Weekly Scrum meetings are per definition open meetings, e.g., everybody from the teams can participate and contribute to the meeting.
  - The weekly meetings are strictly time-boxed.
  - At WP/Project level, the registered committers, contributors, users and adopters are invited to participate
  - At Open ETCS project level, the components of the PMB(Project Management Board) are invited.

- The work-packages resp. tasks need to organize there scrum teams according to practical needs.
- Teams are typically distributed in geography and in organisation (i.e., participating companies).
- Scrum teams typically have to provide several development roles (according to CENELEC and according to Eclipse). Guidance on the possible mixtrues of CENELEC roles into a Scrum team is documented in the appendices section of this guideline.
- To be able to be successful in Agile Development we need to set special focus to the role of the "User" of a product.
  - In general, the user of a product in openETCS should representratives of the project openETCS consuming the result of a scrum team.
  - The workpackage leader of the WP using an outcome of the team is the first candidate.
  - Representatives of partners making use of the openETCS result in long term are also natural users of a team result.
  - Partners in the openETCS project need to agree on the Users before the task when planning the interfaces.
- Each team has to select a scrum master. Scrum training is mandatory.
- A SCRUM master (WP1) is responsible for supporting the teams.
- openETCS project dissemination and exploitation results are managed by WP6
- Software development tasks will be accomplished within the WP2, WP3, WP4, WP5 and WP7

3.1 SCRUM Process in Work Packages

Every work package has its own product backlog. Due to the fact that the openETCS stakeholders are based in different European countries the PB is implemented via the issue tracker on the web based tool on GitHub for arbitrary access . It consists of Use Cases, marked with the label "*US-Operational Scenarios on Utrecht Amsterdam*". Each Use Case has a unique User Story label identifying this Use Case in the backlog. (Sub-) User Stories identified as necessary for the implementation of a Use Case are marked with the user story label.

The actual work at the user stories is broken down in smaller technical tasks. These tasks are organized in the sprint backlog and will be as such labeled. The sprint backlog consists of prioritized tasks and their priorities are integers in the interval [1 . . . 1000], with 1000 representing the highest priority. This priority absolute value is of minor interest as prioritization is only needed to classify backlog items by importance and they are captured in the title of each task

respectively issue. Example: *US-SpeedMonitoring: Dynamic Save Calculation Prio: 500* represents a user story with priority 500. The reason to capture the priority in the title is to make priority changes traceable, as the text of GitHub issues may be changed without leaving any trace.

The labels *Ready*, *In Progress*, and *Done* are used to capture the state of the user stories. While closing of an issue is equivalent to the state *Done* a user story may only be moved to the state *Ready* when:

- a priority has been assigned,
- the scope is clearly defined and distinguished from other user stories in the states *Ready* or *In Progress*,
- and the criteria for reaching the state *Done* are clear.

When setting a user story to *Done* a comment is mandatory for documenting the result. Every project member is eligible to propose user stories. New user stories shall remain in the state *Inbox* until they have been reviewed by the team in a grooming session.

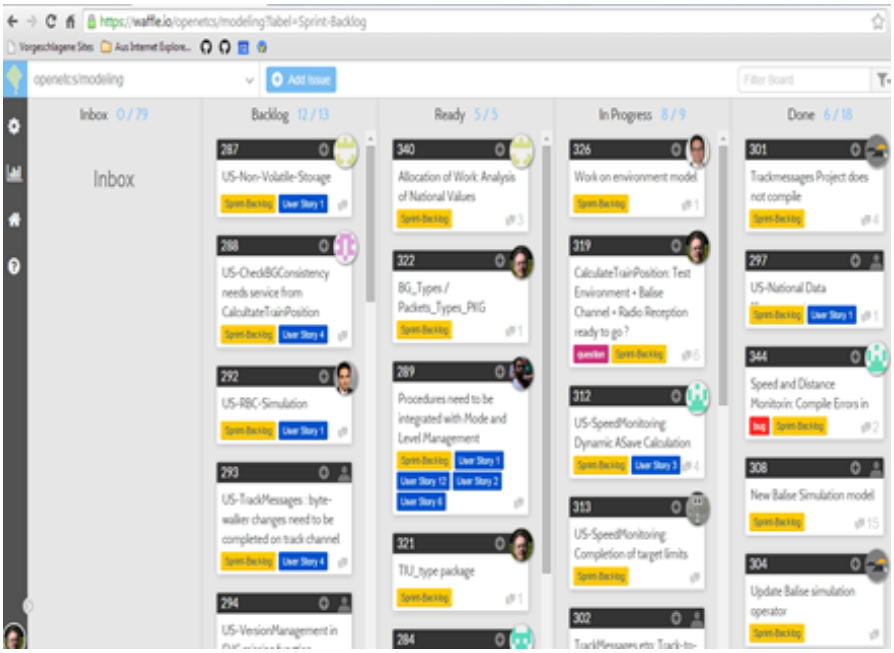


Figure 4. openETCS Backlog in Waffle

The backlog can be visualized as a scrum board with the help of the free web based tool waffle as shown in Figure 4. The use of waffle is completely optional, as it does not provide any additional information. However, waffle provides a more user friendly and intuitive way to display the information in the backlog. When using waffle the backlog items of the same category (e.g. Ready) should be ordered by priority with the highest priority item on top. These priorities have to be captured manually in the title of the issue, i.e. user story, to make it traceable and allow to see priorities without using waffle. By moving backlog items between columns the assignment of the corresponding state tables will automatically take place.

3.2 Scrum of Scrums

As the openETCS organization consists of seven work packages, seven different Scrum teams will need to work together causing interdependencies. SoS is a way to scale and handle these kind of dependency management in big organizations. By applying SoS all WPs can align their interactions and coordinate flow of information between them5.

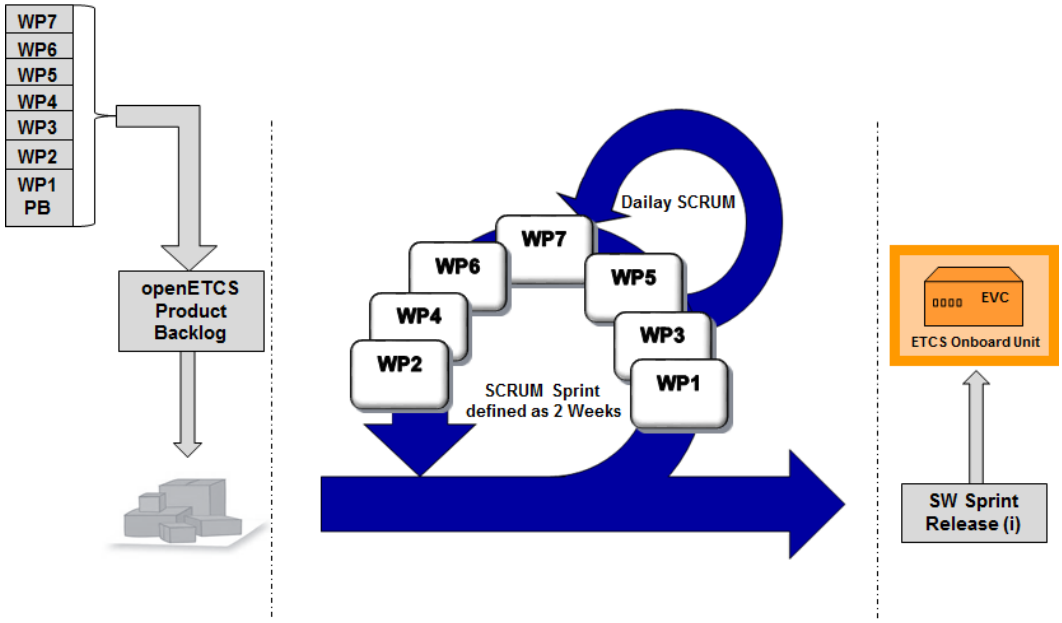


Figure 5. openETCS Scrum of Scrums

openETCS Product Owner (WP1)

The Product Backlog consists of top level *Features* by which the product is described.

- Each Feature itself is estimated by the product owner on how many Story Points this Feature is worth.
- A Feature itself is divided into *Feature Parts*.
- A Feature Part is related directly to a working package and the repository in which the work is documented.
- Each Feature Part itself is estimated and prioritized by the product owner on how many Story Points this Feature Part is worth.

By estimating the feature, the feature becomes accepted on product Level. The Product Backlog is maintained during a regular meeting.

Transferring the Features from the *Product Backlog* to the WP Sprint Backlog During regular grooming sessions the scrum team extracts the items with highest priority out of the Product Backlog and refines them to issues within the repository mentioned within the Feature Part. On these issues Project Poker is performed and by this Story Points are assigned. The decision on which items enter the Sprint Backlog is based on the maximum benefit analysis taking into account the priority as well as the story points.

Sprint Backlog The *Sprint Backlog* consists of issues which are processed during the Sprint. During the Sprint the success is tracked against the Sprint Backlog. The Story Points are earned once the complete result of what has been groomed is uploaded to the GitHub.

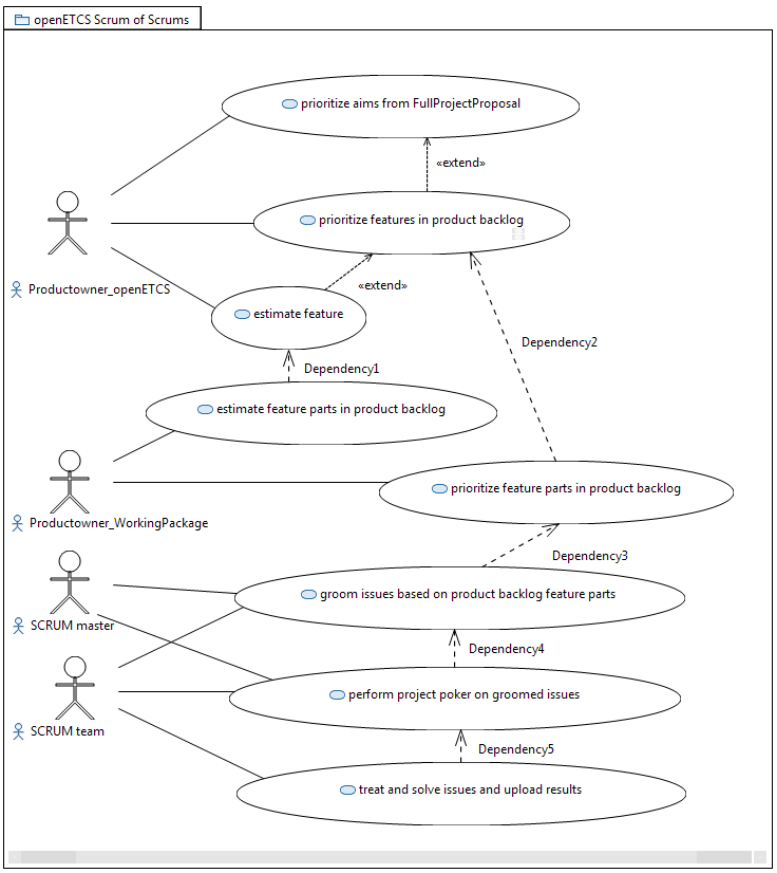


Figure 6. openETCS SoS dependency management

4 Release Principals

The agile SCRUM development has an incremental approach dividing the system into smaller functional parts, which are gradually developed and implemented. The model-based development approach using SCADE makes it possible to incorporate all developed parts as soon as possible and thereby making it possible at an early stage to simulate the provided functionality. This continuous development results in releases with reduced functionality, which are in itself consistent and can be used by other SCRUM Design Teams and the V&V Team to synchronize their functionality or tests.

In this agile development, the incremental delivery quickly allows feedback from the customers, which in the case of openETCS are the project management and other railway stockholders. As an ETCS OBU is a highly safety-relevant system and openETCS is only developing the software kernel the releases cannot be tested in a productive implementation. But the use of simulations and demonstrators allows to address the main functional aspects. In addition, the agile development can combine an external delivery of large components of the overall system with frequent and internal deliveries for simulations. Thereby, the internal releases are not following a concrete pre-planed system, but are triggered by the speed of development and the need of other Scrum Teams to access certain functionalities. The SCRUM Teams select the entries from the backlog which are entered in a sprint and define the criteria when a task is done from their perspective. For example in openETCS a model including a defined set of functions is handed over from the design to the V&V SCRUM Team for independent evaluation and validation of these functions. The V&V team documents it's findings for all artifacts in their respective reports and gives as feedback via GitHub issues back at the Design Team.

Nevertheless, for the overall planning and the interaction with external partners broad milestones are defined in openETCS which are used as a basis for activities which are not performed will modeling the system such as not automatic generated documentation or assessment. This openETCS release interaction between the design and V&V SCRUM sprints is schematically shown in figure 7.

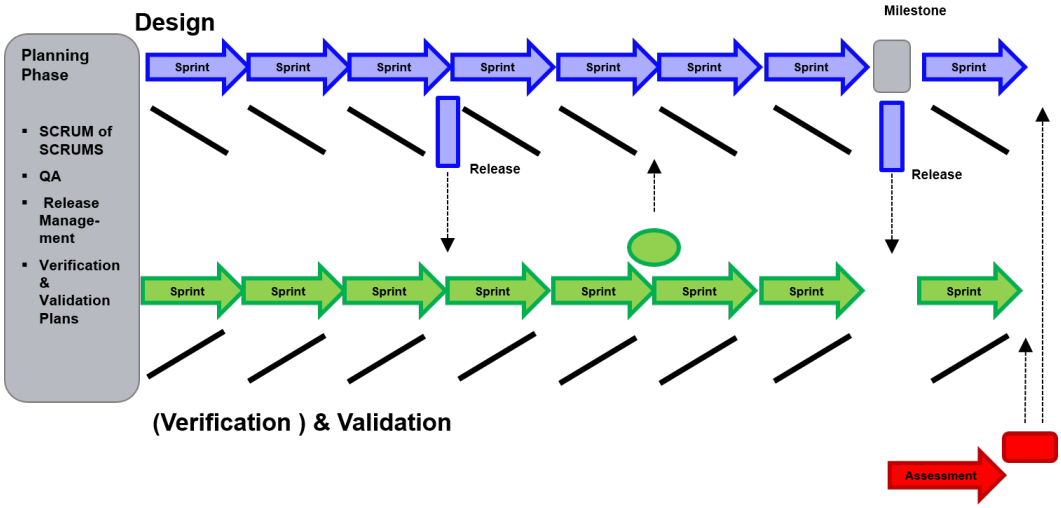


Figure 7. openETCS release concept

As required for SIL 4 development the last step is the assessment of all artifacts for the final release, which have to be reviewed by an authorized independent assessor. This phase cannot be integrated in the continuous agile process. But ideally the assessor is already involved in the main development and thereby reaches preliminary results for long time stable parts of the system. However, the release provided for assessment and afterwards for use has to be a self-sufficient collection which can be reviewed independent of all pre-releases. Since the openETCS project only develops a functional ETCS OBU it is not providing a release ready for a complete safety assessment.