ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

openETCS

openETCS WP3 Modelling Workshop
Requirements Engineering & Traceability

supported 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 CIENCIA E INNOVACIÓN

openETCS@ITEA2 Project

Jan Welte, TU-BS

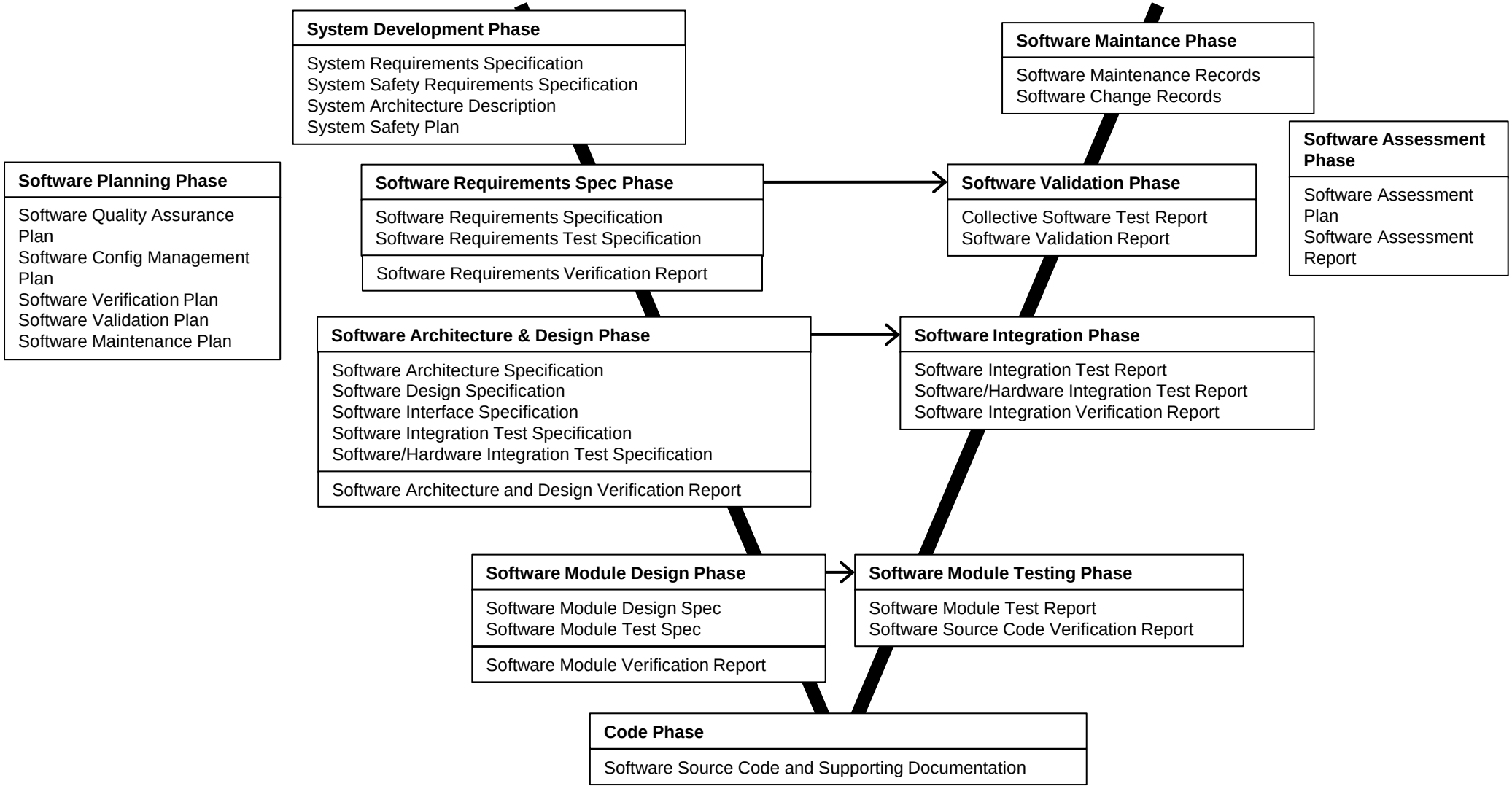
München, 11.03.2014

Agenda

- OpenETCS Software Development
- Traceability
- Requirement Engineering with ProR
- Traceability with ProR

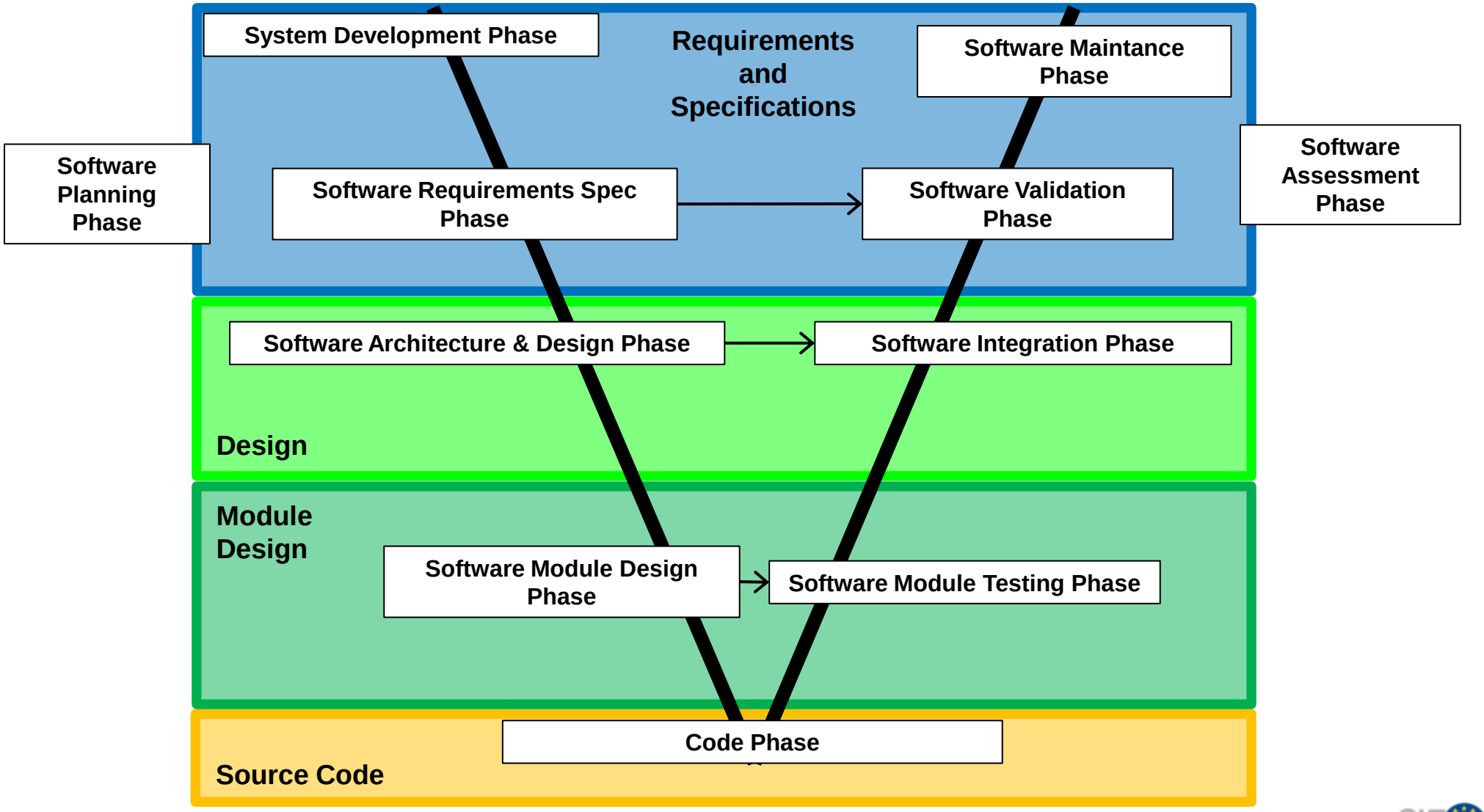
Development Plan

Software Development Lifecycle according to EN 50128



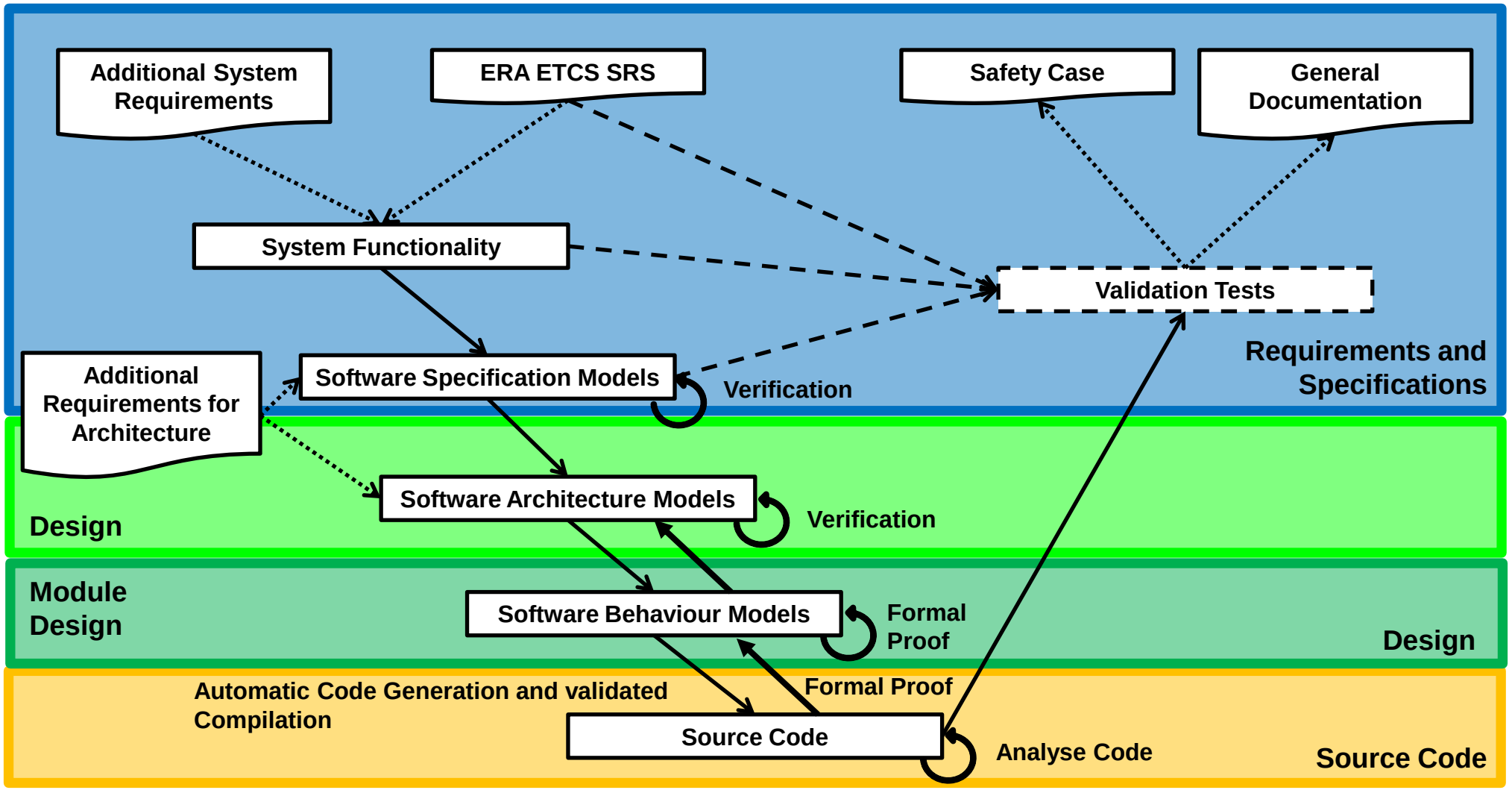
Development Plan

Software Development Lifecycle according to EN 50128



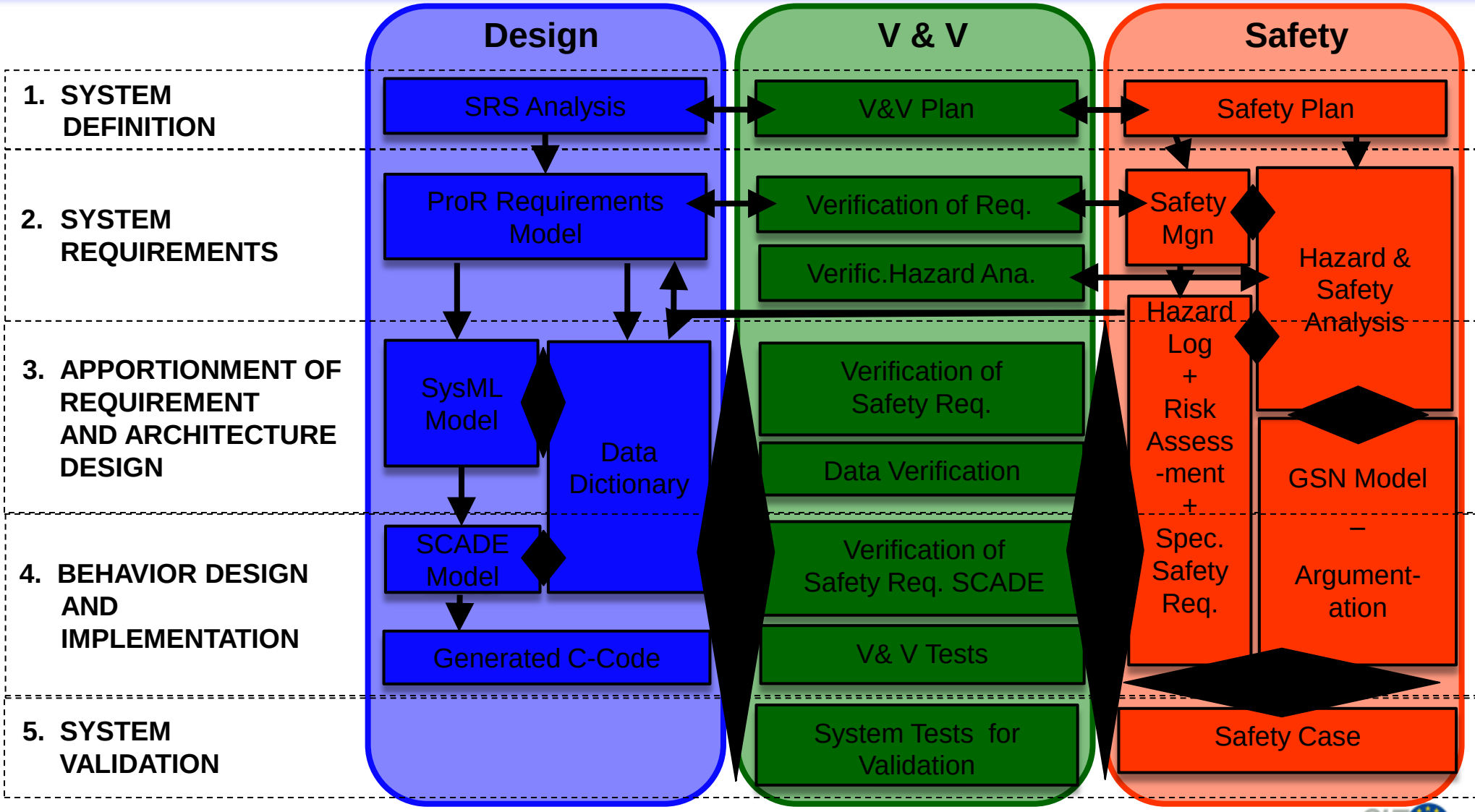
Development Plan

State of the Art Development Process – Formal Proof



Development Plan

OpenETCS Software Development Plan



Development Plan

OpenETCS Software Development Phases

Step	Name	Description	Main Tool component
P1	Software Requirement Phase	Requirement input documents converted to a ReqIF format and informally analyses. The analysis specifies relationships between requirements and revises parts of the requirements to obtain a detail and atomic abstraction level usable for moralization. To support thus the requirements are categorized and grouped.	ProR
P2	Software Architecture Modeling Phase	Building a SysML based on the informal analysis using the categorization of requirements. The architecture model focuses on functional blocks and data flows.	SysML Papyrus
P3	Software Behavior Modeling Phase	Building the SCADE model for the separated basic functional blocks. The SCADE describs the detailed behavior for the function using the data flows.	SCADE
P4	Formal Validation phase	Using test models and model checking techniques, to validate the correct model behavior.	Various

Traceability

The objective of Traceability is to ensure that all requirements can be shown to have been properly met and that no untraceable material has been introduced.

Description

Traceability to requirements shall be an important consideration in the validation of a system and means shall be provided to allow this to be demonstrated throughout all phases of the lifecycle.

Traceability shall be considered applicable to both functional and non-functional requirements and shall particularly address

a) traceability of requirements to the design or other objects which fulfil them,
b) traceability of design objects to the implementation objects which instantiate them,
c) traceability of requirements and design objects to the operational and maintenance objects required to be applied in the safe and proper use of the system,
d) traceability of requirements, design, implementation, operation and maintenance objects, to the verification and test plans and specifications which will determine their acceptability,
e) traceability of verification and test plans and specifications to the test or other reports which record the results of their application.

Traceability

OpenETCS Traceability activities

Traceability Type	Phases	Description	Main Tool Component
Internal Requirement	P1 – P1	Create Links between Requirements	ProR
External Requirement Architecture	P1 – P2	Create Links from Requirement to Architecture Elements in SysML	ProR
Internal Architecture	P2 – P2	Shows Dependencies between SysML Model elements	Probably ReqCycle
External Requirement Behavior	P1 – P3	Create Links between Requirements and SCADE Models	NN
Architecture refinement to Behavior	P2 – P3	Create Dependability between Architecture Elements in SysML and SCADE Models	Probably via SCADE System
External Requirement Validation	P1 – P4	Create Links from Requirement to Validation Test Models in SysML	ProR
Design Verification and Validation	P4 – P2	Create Links from Test Models and Reports to the Architecture Models in SysML	ReqCycle
Behavior Verification and Validation	P4 – P3	Create Links from Test Models and Reports to the Behavior Models in SCADE	Probably via SCADE System

Software Requirement Phase

- Read and understand requirement
- Check whether requirement has to be split or refined, if do so
- create links to related elements
  - Requirements (Derived, Refined, Gathered)
  - Data (SRS Chapter 7 & 8) (Input, Output, Changes)
- Categorize Requirement

Category	Acronym	Description
Note/ Comment	NOT	Notes in the specifications
Architecture	ARC	The requirement introduces some system's modules and describes how they interact
Behavioral	BEH	The requirement describes the steps a particular module perform or the states where the module can be
Data	DAT	The requirement describes the messages some modules exchange
Definitions	DEF	The requirement defines an element or concept in ETCS domain
Environmental	ENV	The requirement describes some constraints/ functionalities on the model coming from the environment of the Kernel
Property	PRP	The requirement describes an expected property of the ETCS
Scenario	SCE	The requirement describes a possible scenario of the ETCS

Traceability with ProR

Software Requirement Phase - Categorization

Category	Acronym	Related Modeling
Note/ Comment	NOT	– Ignore
Architecture	ARC	– Build functional blocks
Behavioral	BEH	– Allocate to a functional block SysML – Model in SCADE
Data	DAT	– Add/ link to Data Dictionary – Allocate to data flow in SysML
Definitions	DEF	– Model block definition diagram – Describe relations between type
Environmental	ENV	– Allocate to specific part of environment
Property	PRP	– Allocaste to functional block – Model as constraints
Scenario	SCE	– Model as sequence diagram if necessary

Formal Mind create an plug-in for ProR to create traceability links

For traceability SysML Proxy elements are created that hold

(1) a reference to the traced Element and

(2) a textual representation of that element (cache).

The actual trace is realized as a Link to the Requirement.

Use Cases

- Create Link (Linking Types: Satisfy, Allocated)
- Change Requirement (not implemented, yet)
- Change Model Element
- Deleting Link
- Deleting Requirement (partially implemented)
- Deleting Element
- Verify (to be defined)

Questions or Discussion

Technische Universität Braunschweig

Institut für Verkehrssicherheit und Automatisierungstechnik iVA

Prof. Dr.-Ing. Dr. h.c. mult. E. Schnieder

Task 4.4 Verification of the tools and processes

Jan Welte

TU Braunschweig

Institute for Traffic Safety and Automation Engineering
welte@iva.ing.tu-bs.de