CRitical S**YST**em Engineering **A**cce**L**eration

Requirement Specification - V1

D202.021

DOCUMENT INFORMATION

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	Requirement Specification - V1
Deliverable No.	D202.021
Dissemination Level	RE
Nature	R
Document Version	V1.00
Date	2014_01_30
Contact	Claudio Pessa
Organization	ALA
Phone	+39 0117562678
E-Mail	claudio.pessa@alenia.it

AUTHORS TABLE

Name	Company	E-Mail
Fabio Bottiglieri	ALA	fabio.bottiglieri@alenia.it
Andrea Casagrande	ALA	andrea.casagrande@alenia.it
Luciana Lo Verde	ALA	Luciana.loverde@alenia.it
Roberto Luisi	ALA	roberto.luisi@alenia.it
Claudio Pessa	ALA	claudio.pessa@alenia.it
Laura Valacca	ALA	laura.valacca@alenia.it
Elena Valfrè	ALA	elena.valfre@alenia.it
Ivo Viglietti	ALA	Ivo.viglietti@alenia.it
Francesco Brunetti	POLITO	francesco.brunetti@polito.it
Giovanni Di Meo	POLITO	giovanni.dimeo@polito.it

REVIEW TABLE

Version	Date	Reviewer
V0.01	2014-01-14	Francesco Brunetti -POLITO
V0.02	2014-01-22	Michael Leute - Cassidian

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected
V0.01	2013-12-05	Version for internal review	ALL
V0.02	2014-01-14	Version after internal review	pag 20-22; 31-32
V1.00	2014-01-30	Version after external review	

CONTENT

1 Introduction

1.1 Role of deliverable

Starting from D202.010 this document has the major scope of detailing the use case description of the Preliminary Design for a new Regional TurboProp (private use case). In particular the main purposes of this deliverable are (from DoW):

- to define the initial version of user requirements including the operational concept for the R&T
- to show the activity sequences to identify all required services that correspond to the operational and support scenarios of the R&T;
- to show the operational scenarios to describe the interactions between users, tools and services.

1.2 Relationship to other CRYSTAL Documents

This deliverable collects inputs directly from CRYSTAL D202.010 "Use Case Description" because can be considered the detailed refinement of it. It has a strong relationship with D202.031, while D202.021 is more focused in describing the requirements from the user point of view while D202.031 is more focused in describing the envisaged IT environment (SEE) enabling those user requirements. Moreover this deliverable tries to follow a common approach in describing the assets and requirements with the other "user requirement definition" deliverables under development in SP2.

1.3 Structure of this document

This document is composed of 4 main chapters:

- Chapter 1 gives an overview of the scope of the deliverable, relationship with other CRYSTAL documents and this description of the document structure
- Chapter 2 describes the overall approach of and give a "data model" of the information used to describe the scenarios and the use case and maps the relationship with D202.031
- Chapter 3 instantiates the described approach for each scenario described in D202.010
- Chapter 4 refines the description of the engineering methods by describing them as sequence of Engineering Activities, pre-conditions, post-conditions and Artefacts for each phase.

2 Overall Approach

2.1 WP202 data model

In order to avoid misunderstanding a data model of this approach has been prepared: it contains the objects used in the description of WP202 and the relationship among them. In the following schema the boundaries of the deliverables D202.021 in red and D202.031 in green have been pointed out.

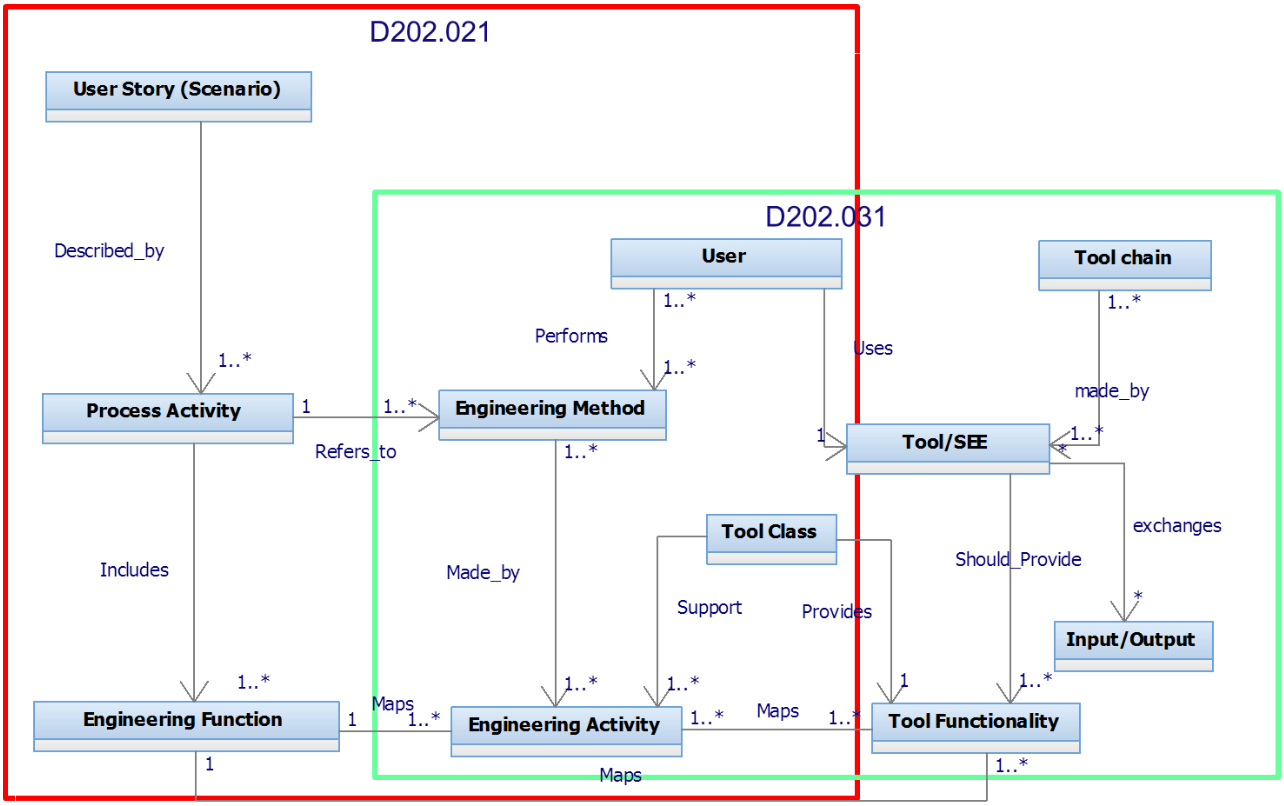


Figure 1: Approach data model and deliverables boundaries

Here below a short description of the objects shown in Figure 1

- User stories: the scenarios described in D202.010
- Process Activities: the main macro-activities that describe what to do in order to carry out the User Stories
- Engineering Function: each function that compose the flow chart or sequence that describe the macro–activity
- Engineering Method: it describes how the process activity is performed
- Engineering Activity: it describes each step of the Engineering Method
- Tool Class: the kind of tool that should support an engineering activity by providing a tool functionality

- User: who is carrying out the engineering method
- Tool/SEE: the instantiation of the tool class that actually is used to perform the engineering activity
- Tool Functionality: the functionality given by the tool used to perform an engineering activity
- Tool chain: the tools used to perform a User story considering the link among them and the sequence in which they are used
- Input/Output: what is exchanged among the tools in the tool chain

The following steps have been made to gather user requirements:

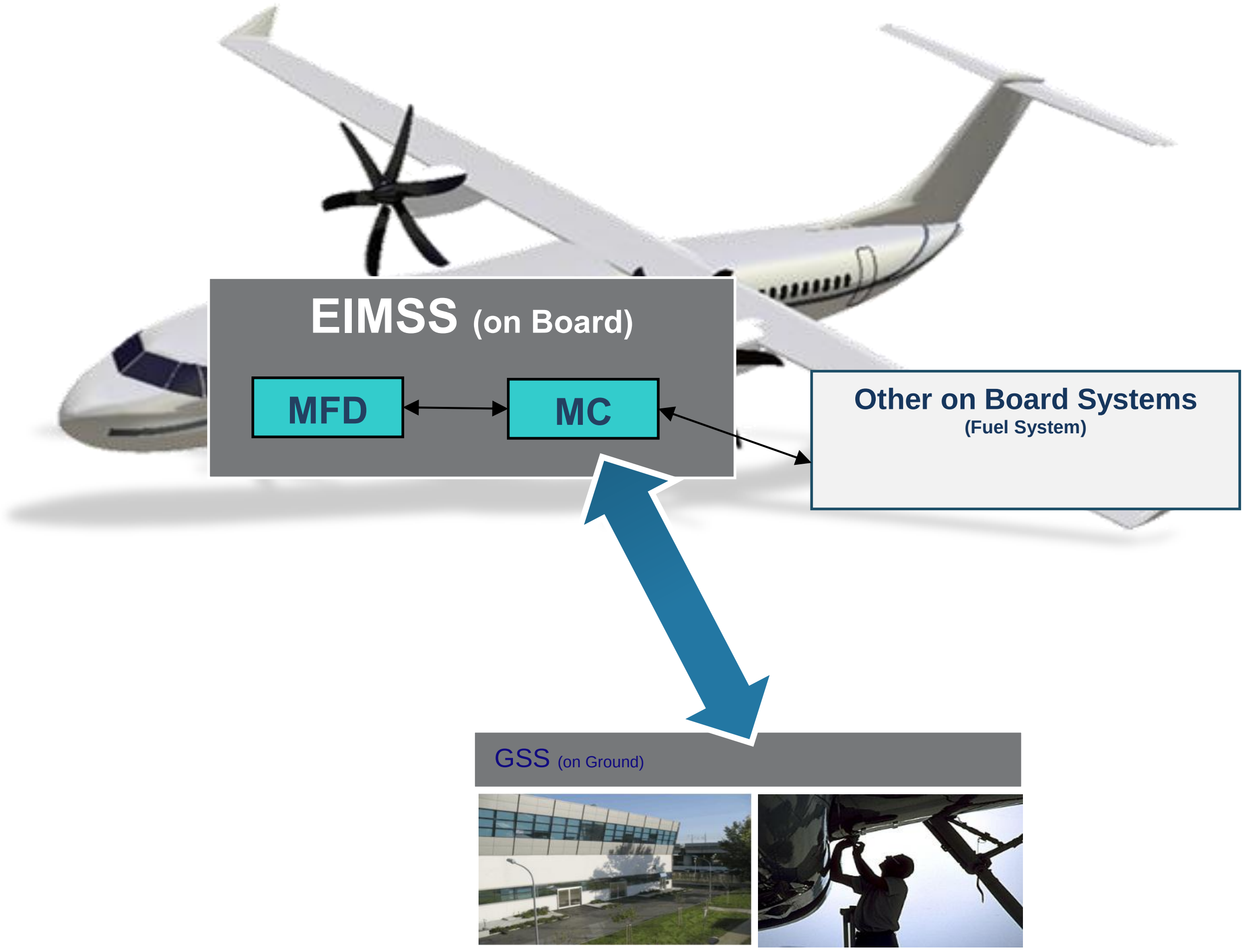
1. To explicit the Process Activities that compose the User Story described in D202.010 scenarios
2. To describe Process Activities as flow charts or sequence of Engineering Function
3. To link Process Activities to Engineering Methods (EM) and Users and Tool Class
4. To link EM with Engineering Activity (IT Capability - what we expect a class of tools should be able to do in order to satisfy EM)

3 Scenarios Description

In the following sub-chapters the scenarios will be described by the instantiation of the first three steps of the process described in chapter 2, while the fourth step (link between Engineering method and engineering activity) will be done in chapter 4. The scenarios are capturing design process key points for the "Preliminary Design for a new Regional TurboProp" use case. The use case has been widely described in D202.010, here a short reminder:

The WP202 use case is focused on the description of the preliminary definition and design of an Enhanced Integrated Monitoring and Support System (EIMSS) for a new turboprop regional aircraft. The main objectives of this advanced system are:

- Support pilot operations by providing data in real time
- Support maintenance base operators by providing a subset of systems data and pilot's actions before the aircraft is landed in order to anticipate maintenance decisions and then minimize the down time of the system.
- Support maintenance ground operators with Off-line systems data availability



The envisaged scenarios are the following:

Scenario 1 – Functional Model Development

Scenario 2 – Functional Model Analysis

Scenario 3 – Functional View under Configuration Control

Scenario 1b – Physical Model Development

These scenarios have been already described in D202.010 and there they have been refined and matured, where in the following chapters there are changes from what described in the previous deliverable, it will be underlined and substantiated.

The “optional” scenario Scenario 2b – Physical Model Analysis will be further defined in version V2 (M20) of this deliverable.

3.1 Scenario 1 – Functional Model Development

This scenario aims to describe the development of a functional model starting from a requirement baseline considering the several intermediate steps to get the animation of the model in order to evaluate the functionalities expected by the different stakeholders against the requirement baseline.

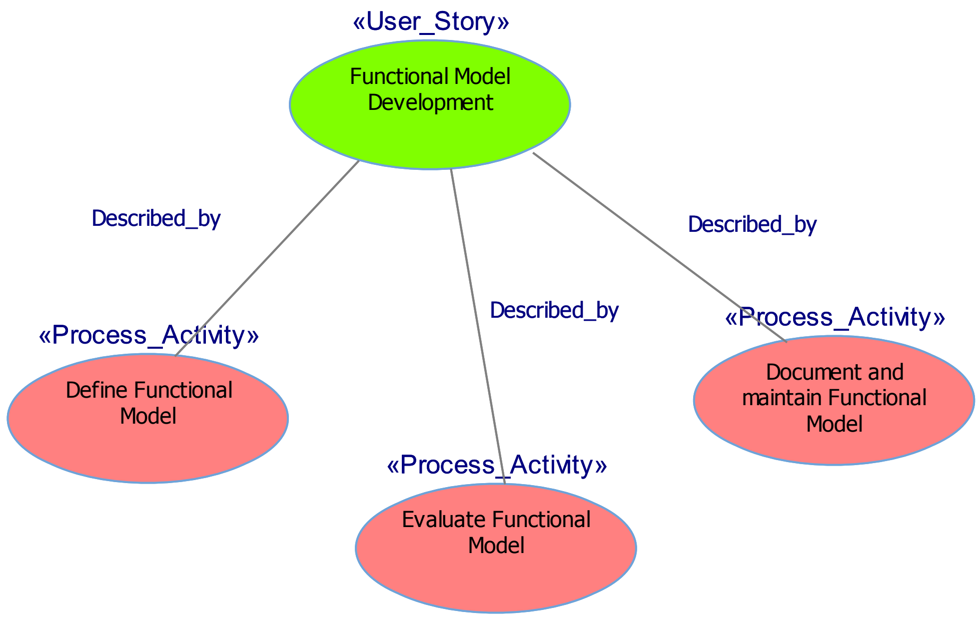


Figure 2: "Functional Model Development" user story

The User Story is described by three main Process Activities:

- Define Functional Model
- Evaluate Functional Model
- Document and maintain Functional Model

3.1.1 Define Functional Model

This process activity is described by the following Engineering Functions:

1. To define functional requirements
2. To analyse functional requirements
3. To allocate functional requirements to scenarios
4. To define the flow of activities
5. To define the sequence of activities
6. To allocate activities to system elements
7. To identify the interfaces between system elements
8. To define the input/output of system elements

	Engineering Functions	Engineering Method	Tool Class	User
1	To define functional requirements	Analyse Requirement	Requirement Manager	System Engineer
2	To analyse functional requirements	Analyse Requirement	Requirement Manager	System Engineer
3	To allocate functional requirements to scenarios	Verify Design against Requirements	Functional Modeller	System Engineer
4	To define the flow of activities	Verify Design against Requirements	Functional Modeller	System Engineer
5	To define the sequence of activities	Verify Design against Requirements	Functional Modeller	System Engineer
6	To allocate activities to system elements	Verify Design against Requirements	Functional Modeller	System Engineer
7	To define the input/output of system elements	Verify Design against Requirements	Functional Modeller	System Engineer

Table 1: Scenario 1 - Engineering Functions links – Define Functional Model

3.1.2 Evaluate Functional Model

This process activity is described by the following Engineering Functions:

1. To define system functional model alternatives
2. To define evaluation criteria (requirement & constraint)
3. To measure system model ability to satisfy evaluation criteria
4. To select and refine functional model

	Engineering Functions	Engineering Method	Tool Class	User
1	To define system functional model alternatives	Verify Design against Requirements	Requirement Manager	System Engineer
2	To define evaluation criteria (requirement & constraint)	Analyse Requirement	Requirement Manager	System Engineer
3	To measure system model ability to satisfy evaluation criteria	Verify Design against Requirements	Functional Modeller	System Engineer
4	To select and refine functional model	Verify Design against Requirements	Functional Modeller	System Engineer

Table 2: Scenario 1 - Engineering Functions links - Evaluate Functional Model

3.1.3 Document and maintain Functional Model

This process activity is described by the following Engineering Functions:

1. To document the functional model
2. To document relevant decision on model
3. To establish traceability with requirement and solution
4. To manage model configuration and model baselines

	Engineering Functions	Engineering Method	Tool Class	User
1	To document the functional model	Verify Design against Requirements	Functional Modeller	System Engineer
2	To document relevant decision on model	Analyse Requirement	Functional Modeller	System Engineer
3	To establish traceability with requirement and solution	Verify Design against Requirements	Requirement Manager, Product Lifecycle Manager	System Engineer Configuration Manager
4	To manage model configuration and model baselines	Verify Design against Requirements	Product Lifecycle Manager	Configuration Manager

Table 3: Scenario 1 - Engineering Functions links - Document and Maintain Functional model

3.2 Scenario 2 – Functional Model Analysis

This scenario aims to describe the several steps to be performed in order to analyse the functional model and verify its consistency to the system functional requirements. In particular, here the focus is on reliability and testability analysis of the system to be performed through specific disciplinary tool class.

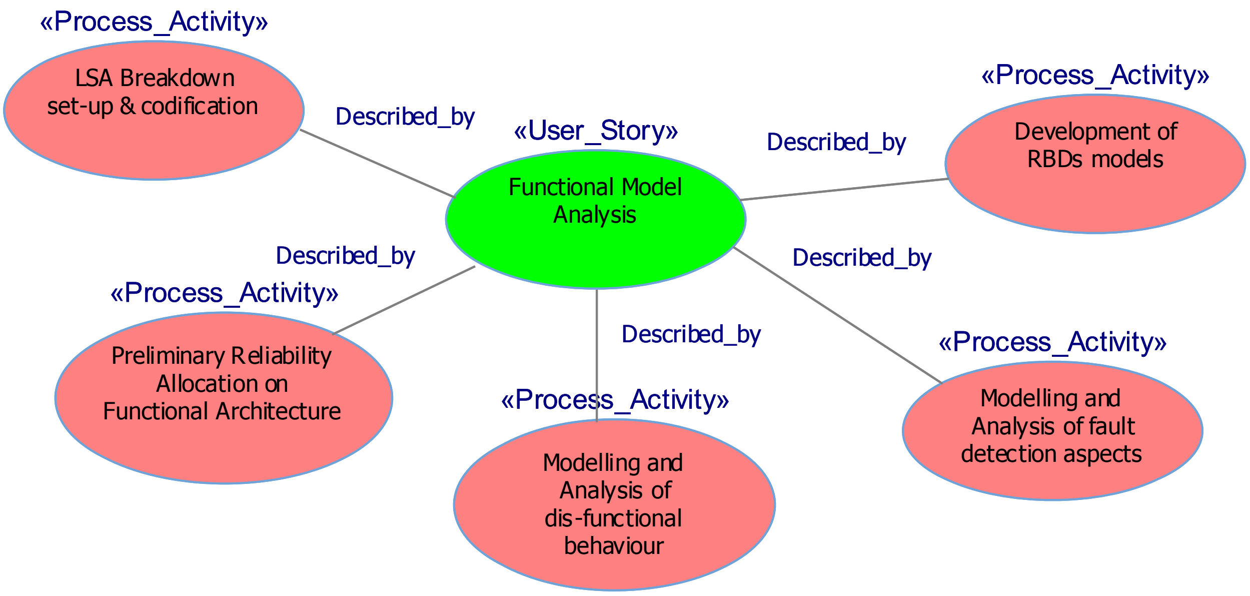


Figure 3: "Functional Model Analysis" user story

3.2.1 LSA Breakdown set-up & codification

This process activity is described by the following Engineering Functions:

1. To model specific LSA view-point of Functional Architecture of Fuel System
2. To organize Functional Architecture Baseline of Fuel System as ASD S1000D coded Breakdown
3. To publish Fuel System Functional Architecture codification
4. To release and share Functional Architectural codification in functional view

	Engineering Functions	Engineering Method	Tool Class	User
1	To model Functional Architecture of Fuel System	Verify Design against Requirements	Functional Modeller	System Engineer
2	To organize Functional Architecture Baseline of Fuel System as ASD S1000D coded Breakdown	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
3	To publish Fuel System Functional Architecture codification	Verify Design against Requirements	Functional Modeller	RM&T Engineer
4	To release and share Functional Architectural codification in functional view	Verify Design against Requirements	Product Lifecycle Manager	RM&T Engineer

Table 4: Scenario 2 - Engineering Functions links - LSA Breakdown set-up & codification

3.2.2 Preliminary Reliability Allocation on Functional Architecture

This process activity is described by the following Engineering Functions:

1. To add LSA information on Functional Architecture of Fuel System
2. To allocate reliability figure to basic elements (BDD,IBD, parts) of Fuel System Functional Architecture
3. To evaluate preliminary Fuel System Reliability parameter (e.g. MTBF)
4. To publish Reliability Result as MOE of Reliability performance System requirement

	Engineering Functions	Engineering Method	Tool Class	User
1	To describe Functional Architecture of Fuel System	Verify Design against Requirements	Functional modeller	System Engineer
2	To allocate reliability figure to basic elements (blocks, parts) of Fuel System Functional Architecture	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
3	To evaluate preliminary Fuel System Reliability parameter (e.g. MTBF)	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
4	To publish Reliability Result as MOE of Reliability performance System requirement	Verify Design against Requirements	Requirement Manager	RM&T Engineer

Table 5: Scenario 2 - Engineering Functions links - Preliminary reliability Allocation on functional Architecture

3.2.3 Modelling and Analysis of dis-functional behaviour

This process activity is described by the following Engineering Functions:

1. System Engineer releases LSA view-point for Functional Architecture of Fuel System in Functional Modeller
2. RM&T Engineer identifies main functional deviation (failure modes) on basic elements (BDD,IBD, parts) of Functional Architecture in RM&T Analysis tool (FMEA Module)
3. RM&T Engineer releases functional deviation as additional attributes to basic elements (BDD,IBD, parts) of Functional Architecture in Functional Modeller.
4. System Engineer integrates dis-functional behaviour on Fuel System Functional Architecture (Use case, Behaviour, Scenario models) in Functional Modeller

5. RM&T Engineer performs simulation of single failure scenario in Functional Modeller.
6. RM&T Engineer renames each single failure scenario with Failure Mode Id (FMI)
7. RM&T Engineer releases main result of each failure mode scenario (intermediate/final parameters / states) to RM&T Analysis tool (FMEA Module)
8. RM&T Engineer fill single failure effects in FMEA table within PTC Windchill QS

	Engineering Functions	Engineering Method	Tool Class	User
1	To release Functional Architecture of Fuel System in Functional Modeller	Verify Design against Requirements	Functional modeller	System Engineer
2	To identify main functional deviation (failure modes) on basic elements (BDD,IBD, parts) of Functional Architecture in RM&T Analysis tool (FMEA Module)	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
3	To release functional deviation as additional attributes to basic elements (BDD,IBD, parts) of Functional Architecture in Functional Modeller.	Verify Design against Requirements	Functional modeller	RM&T Engineer
4	To integrates dis-functional behaviour on Fuel System Functional Architecture (Use case, Behaviour, Scenario models) in Functional Modeller	Verify Design against Requirements	Functional modeller	System Engineer
5	To perform simulation of single failure scenario in Functional Modeller.	Verify Design against Requirements	Functional modeller	RM&T Engineer
6	To rename each single failure scenario with Failure Mode Id (FMI)	Verify Design against Requirements	Functional modeller	RM&T Engineer
7	To release main result of each failure mode scenario (intermediate/final parameters / states) to RM&T Analysis tool (FMEA Module)	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
8	To fill single failure effects in FMEA table within RM&T Analysis tool	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer

Table 6: Scenario 2 - Engineering Functions links - Modelling and Analysis of dis of dis-functional behaviour

3.2.4 Modelling and Analysis of fault detection aspects

This process activity is described by the following Engineering Functions:

1. System and RM&T Engineer release Functional Architecture (including dis-functional behaviour) of Fuel System in Functional Modeller
2. System and RM&T Engineer integrate Functional Architecture Baseline (including dis-functional behaviour) of Fuel System with Fault detection capability model (Use case, Behaviour, Scenario models) in Functional Modeller
3. RM&T Engineer updates simulation of single failure scenario in Functional Modeller.
4. RM&T Engineer releases main result of each failure mode scenario (intermediate/final parameters / states) to RM&T Analysis tool (FMEA Module)
5. RM&T Engineer fills single failure detection coverage in FMEA table within PTC Windchill QS.

6. RM&T Engineer publishes Testability Result in Requirement Manager as MOE of System Testability performance requirement

	Engineering Functions	Engineering Method	Tool Class	User
1	To release Functional Architecture (including dis-functional behaviour) of Fuel System	Verify Design against Requirements	Functional modeller	System Engineer, RM&T Engineer
2	To integrate Functional Architecture Baseline (including dis-functional behaviour) of Fuel System with Fault detection capability model (Use case, Behaviour, Scenario models)	Verify Design against Requirements	Functional modeller	System Engineer, RM&T Engineer
3	RM&T Engineer updates simulation of single failure scenario in Functional Modeller.	Verify Design against Requirements	Functional modeller	RM&T Engineer
4	To release main result of each failure mode scenario (intermediate/final parameters / states) (FMEA Module)	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
5	To fill single failure detection coverage in FMEA table	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
6	To publish Testability Result as MOE of System Testability performance requirement	Verify Design against Requirements	Requirement Manager	RM&T Engineer

Table 7: Scenario 2 - Engineering Functions links - Modelling and Analysis of fault detection aspects

3.2.5 Development of RBDs models

This process activity is described by the following Engineering Functions:

1. System and RM&T Engineer release Functional Architecture (including dis-functional behaviour) of Fuel System in Functional Modeller
2. System and RM&T Engineer integrate Mission Abort condition in Functional Architecture (including dis-functional behaviour) of Fuel System in Functional Modeller (Behaviour, Scenario models)
3. RM&T Engineer develops RBD models relevant to Mission Abort condition in Functional Architecture (including dis-functional behaviour) of Fuel System in Functional Modeller (safety profile package)
4. RM&T Engineer publishes RBD (Basic block elements and logic ports) to RM&T Analysis tool (RBD Module)
5. RM&T Engineer allocates reliability figure to basic block elements of RBD in RM&T Analysis tool (RBD Module)
6. RM&T Engineer publish Mission Reliability Result in Requirement Manager as MOE of System Mission reliability performance requirement

	Engineering Functions	Engineering Method	Tool Class	User
1	System and RM&T Engineer release Functional Architecture (including dis-functional behaviour) of Fuel System in Functional Modeller	Verify Design against Requirements	Functional modeller	System Engineer, RM&T Engineer
2	System and RM&T Engineer integrate Mission Abort condition in Functional Architecture (including dis-functional behaviour) of Fuel System in Functional Modeller (Behaviour, Scenario models)	Verify Design against Requirements	Functional modeller	System Engineer, RM&T Engineer
3	RM&T Engineer develops RBD models relevant to Mission Abort condition in Functional Architecture (including dis-functional behaviour) of Fuel System in Functional Modeller (safety profile package)	Verify Design against Requirements	Functional modeller	RM&T Engineer
4	RM&T Engineer publishes RBD (Basic block elements and logic ports) to RM&T Analysis tool (RBD Module)	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
5	RM&T Engineer allocates reliability figure to basic block elements of RBD in RM&T Analysis tool (RBD Module)	Verify Design against Requirements	RM&T Analysis tool	RM&T Engineer
6	RM&T Engineer publish Mission Reliability Result in Requirement Manager as MOE of System Mission reliability performance requirement	Verify Design against Requirements	Requirement Manager	RM&T Engineer

Table 8: Scenario 2 - Engineering Functions links - Development of RBDs models

3.3 Scenario 3 – Functional View under Configuration Control

This scenario aims to investigate the possibility to manage the items related with the functional view that are typically generated during the conceptual and preliminary phases of the design of an aeronautical product and that, till now, are not strongly linked with the items generated downstream in the project and that populate the “as designed”, “as planned”, etc. views.

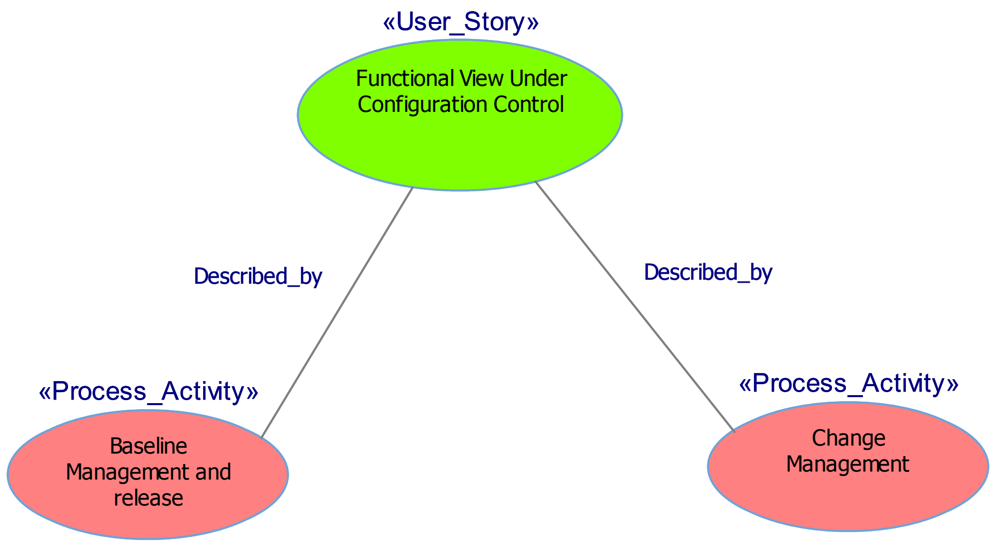


Figure 4: "Functional view under Configuration Control" user story

The User Story is described by two main Process Activities:

- Baseline Management and release
- Change Management

3.3.1 Baseline Management and release

This process activity is described by the following Engineering Functions:

1. To identify the set of Requirements for FUEL System
2. To create a Requirement baseline
3. To control the Requirement baseline
4. To configure the Requirement Baseline in As-Required view
5. To analyse and develop FUEL System Functional Requirements
6. To release Functional model Baseline
7. To analyse and develop FUEL System Functional & Performance Requirements
8. To release functional model baseline
9. To control the Functional baselines
10. To configure the Functional model Baseline in As-Conceived view

In the following table are shown the Engineering Functions links with Engineering Methods, Users and Tool Class.

	Engineering Functions	Engineering Method	Tool Class	User
1	To identify the set of Requirements for FUEL System	Analyse Requirement	Requirement Manager	System Engineer
2	To create a Requirement baseline	Provide configuration Control	Requirement Manager	System Engineer
3	To control the Requirement baseline	Provide configuration Control	Requirement Manager	System Engineer
4	To configure the Requirement Baseline in As-Required view	Provide configuration Control	Product Lifecycle Manager	Configuration Manager
5	To analyse and develop FUEL System Functional Requirements	Analyse Requirement, Verify Design against Requirements	Functional Modeller	System Engineer
6	To release Functional model Baseline	Provide configuration Control	Product Lifecycle Manager	Configuration Manager
7	To analyse and develop FUEL System Functional & Performance Requirements	Analyse Requirement, Verify Design against Requirements	Functional Modeller	System Engineer
8	To release functional model baseline	Provide configuration Control	Product Lifecycle Manager	Configuration Manager
9	To control the Functional baselines	Provide configuration Control	Functional Modeller	System Engineer
10	To configure the Functional model Baseline in As-Conceived view	Provide configuration Control	Product Lifecycle Manager	Configuration Manager

Table 9: Scenario 3 - Engineering Functions links – Baseline Management and release

3.3.2 Change Management

This process activity is described by the following Engineering Functions:

1. To manage change proposal
2. To set the boundaries of the change though the different configuration view
3. To summon a change board and involve domain experts and stakeholders
4. To assess the impacts of the change through the different views
5. To define the new requirements that drive the change
6. To analyse all functional artefacts correlated to the new requirements.
7. To deploy the change in the functional model
8. To document the deployment of the change
9. To control and release all the Configuration Items impacted by the change

	Engineering Functions	Engineering Method	Tool Class	User
1	To manage change proposal	Provide Process Management	Workflow Manager	System Engineer
2	To set the boundaries of the change though the different configuration view	Provide configuration Control	Requirement Manager	Configuration Manager
3	To summon a change board and involve domain experts and stakeholders	Provide Process Management	Workflow Manager	System Engineer
4	To assess the impacts of the change through the different views	Provide configuration Control	Product Lifecycle Manager	Configuration Manager
5	To define the new requirements that drive the change	Analyse Requirement,	Requirement Manager	System Engineer
6	To analyse all functional artefacts correlated to the new requirements.	Verify Design against Requirements	Functional Modeller	System Engineer
7	To deploy the change in the functional model	Verify Design against Requirements	Functional Modeller	System Engineer
8	To document the deployment of the change	Verify Design against Requirements	Product Lifecycle Manager	System Engineer
9	To control and release all the Configuration Items impacted by the change	Provide configuration Control	Functional Modeller	System Engineer

Table 10: Scenario 3 - Engineering Functions links - Change Management

3.4 Scenario 1b – Physical Model Development

This scenario aims to investigate how to link the development of the physical model of a fuel system to the former (in development lifecycle) functional model considering how the interfaces between functional elements will be instantiated in the physical interfaces between "real" system element. Then assess the physical model against performance requirement for preliminary verification.

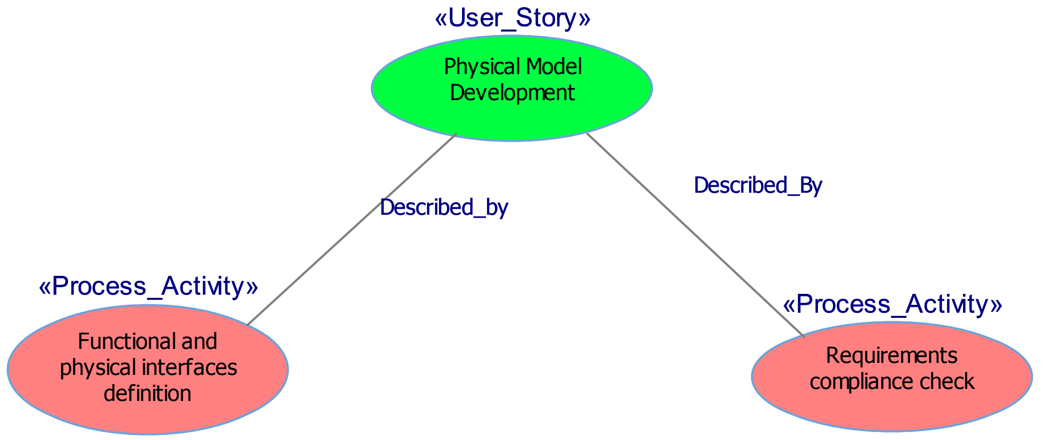


Figure 5 : "Physical Model Development" user story

The User Story is described by two main Process Activities:

- Functional and physical interfaces definition
- Requirements compliance check

3.4.1 Functional and physical interfaces definition

This process activity is described by the following Engineering Functions:

1. To identify the set of Requirements for the FUEL system
2. To identify the Functional model of the FUEL system
3. To identify the architecture of the FUEL system
4. To analyze the architecture and main components of the FUEL system
5. To analyze component's physical parameters of the FUEL system
6. To release physical model of the FUEL system
7. To release relevant physical parameters (and their ranges) to be monitored
8. To link relevant physical parameters as interfaces between functional views of EIMSS and FUEL system

In the following table are shown the Engineering Functions links with Engineering Methods, Users and Tool Class.

	Engineering Functions	Engineering Method	Tool Class	User
1	To identify the set of Requirements for FUEL System	Analyse Requirement	Requirement Manager	Analyst Engineer
2	To identify the Functional model of the FUEL system	Heterogeneous Simulation, Maintain Consistency between multi-viewpoint models	Functional Modelling	Analyst Engineer
3	To identify the architecture of the FUEL system	Heterogeneous Simulation, Maintain Consistency between multi-viewpoint models	Functional Modelling	Analyst Engineer
4	To analyse the architecture and main components of the FUEL system	Heterogeneous Simulation	Physical Modelling	Analyst Engineer
5	To analyse component's physical parameters of the FUEL system	Heterogeneous Simulation	Physical Modelling	Analyst Engineer
6	To release physical model of the FUEL system	Heterogeneous Simulation	Physical Modelling	Analyst Engineer
7	To release relevant physical parameters (and their ranges) to be monitored	Heterogeneous Simulation	Physical Modelling	Analyst Engineer
8	To link relevant physical parameters as interfaces between functional views of EIMSS and FUEL system	Heterogeneous Simulation, Maintain Consistency between multi-viewpoint model	Functional and Physical Modelling	System Engineer, Analyst Engineer

3.4.2 Requirements compliance check

This process activity is described by the following Engineering Functions:

1. To identify the set of requirements for the FUEL system
2. To identify the physical model of the FUEL system
3. To identify the physical parameters and their values
4. To identify physical parameters within requirements
5. To verify the Means of Compliance of physical parameters of requirements
6. To release compliance evidences or validation

In the following table are shown the Engineering Functions links with Engineering Methods, Users and Tool Class.

	Engineering Functions	Engineering Method	Tool Class	User
1	To identify the set of requirements for the FUEL system	Analyse Requirement	Requirement Manager	Analyst Engineer
2	To identify the physical model of the FUEL system	Heterogeneous Simulation	Physical Modelling	Analyst Engineer
3	To identify the physical parameters and their values	Heterogeneous Simulation	Physical Modelling	Analyst Engineer
4	To identify physical parameters within requirements	Analyse Requirement	Requirement Manager	Analyst Engineer
5	To verify the Means of Compliance of physical parameters of requirements	Analyse Requirement, Verify Design against Requirements	Functional Modelling , Requirement Manager	Analyst Engineer
6	To release compliance evidences or validation	Analyse Requirement, Verify Design against Requirements	Functional Modelling	Analyst Engineer

4 Engineering methods and Engineering Activities

This has chapter has the main goal of describing the Engineering Methods used in the scenarios considering:

- Engineering Activities in which the Engineering Method is decomposed
- Pre-condition and Post-condition for the Engineering Method
- Artefact used in the Engineering Methods and Artefacts as input /output for the Engineering Method

The Engineering methods here described have commonalities with the ones from the Public Use case but can differ from them considering the different environment in which they are used.

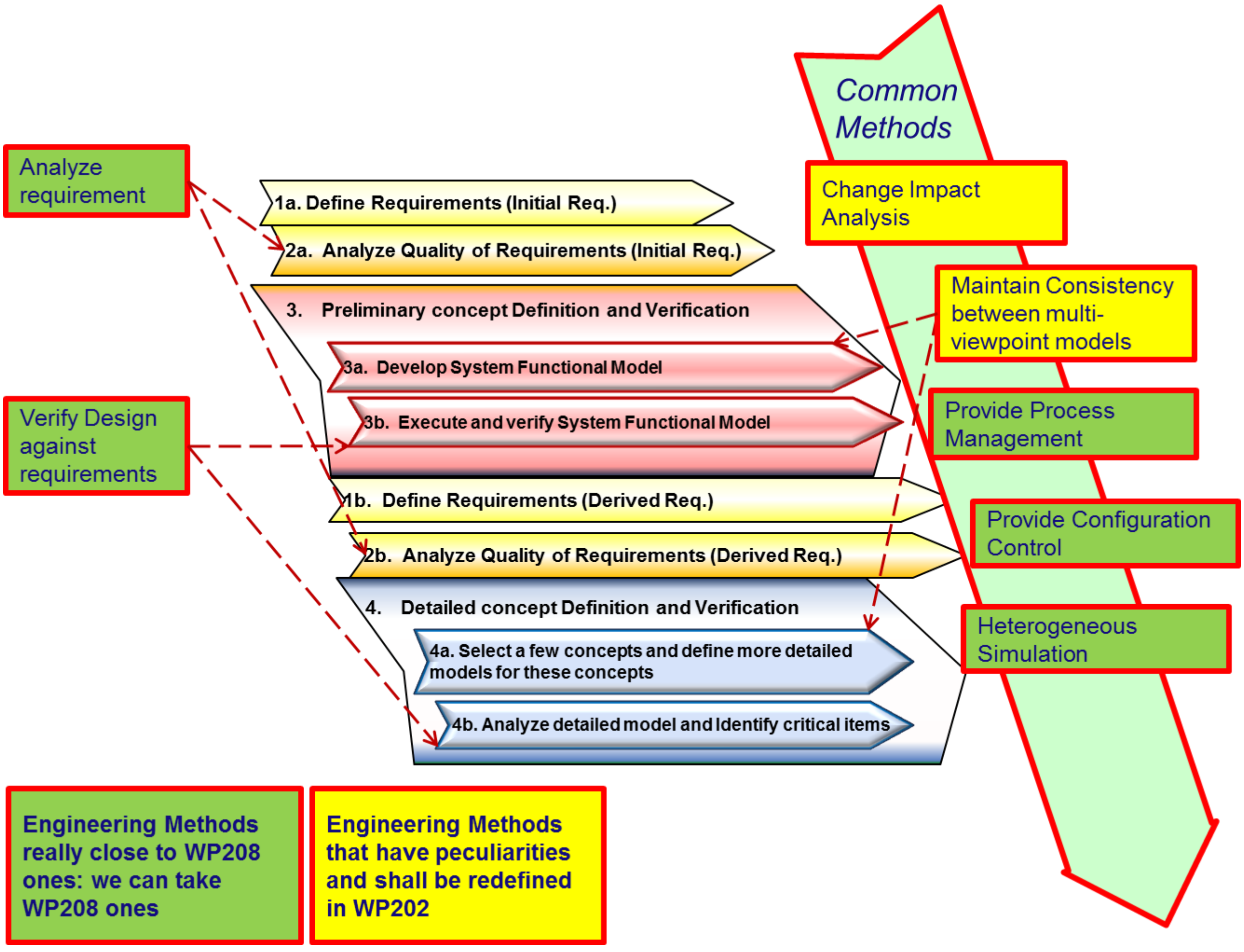


Figure 6: Engineering methods through the Preliminary Design of a new Regional Turboprop

These Engineering Methods, already described in the annex of D202.010, have been further refined according the increased maturity and knowledge of the envisaged use case.

4.1 Analyse Requirements

Engineering Method: Analyse Requirement		
Purpose: to provide an engineered method to evaluate quality, consistency and coverage of the system / subsystems requirements		
Pre-Condition Requirement is stored in Requirements manager D/B Requirement quality information is stored in other rqa D/B	**Engineering Activity** 1. Structure and categorize requirements in requirement manager 2. Establish quality policy for requirement types in requirement quality analyser 3. Requirement quality analyser send request for requirement list import to Requirement Manager Tool 4. Requirement Manager tool assembles the list of requested system requirements and send back to rqa tool 5. rqa tool receives requirements 6. in rqa tool requirements list are automatically analysed 7. change needs arise from rqa, new requirement proposal are sent back to requirement manager 8. in requirement manager are evaluated new requirement impact on overall requirement structure	**Post-Condition** Requirements quality, and consistency are checked and evaluated, a new baseline list of requirement results are traced according the degree of formalization (formal review baseline, peer review baseline, working baseline)
Artefacts provided as input of the activity 1. requirement list 2. quality metrics	**Artefacts** produced during the activity 1. requirement list 2. requirement change proposal 3. list of impacted requirement	**Artefacts** which are the result of the activity 1. new requirement list 2. evaluation report (metric)
Additional Comments: N/A		

4.2 Verify Design against Requirements

Engineering Method: Verify Design against Requirements

Purpose: The aim of this method is to verify via different CAE solutions, at different level of detail and from different viewpoints (functionalities, performances, -abilities, etc.) the compliance of the solution against the requirements.

Pre-Condition	**Engineering Activity**	**Post-Condition**
Requirement list available functional model available physical model available	1. In RM&T Analysis Tool launch service «Get list of System Functional Architectural model basic elements (Use Case, Activities, Blocks, Parts). 2. In RM&T Analysis Tool modeller Functional Architectural elements are organised in Logistic Breakdown and ASD S1000D reference code is assigned to breakdown elements 3. In RM&T Analysis Tool modeller preliminary reliability figures are allocated to basic System functional architectural model elements. 4. In RM&T Analysis Tool modeller, main functional deviation (failure modes) are identified on basic System functional architectural model elements. 5. In the functional modeller, launch service «Get list of System Functional Architectural model basic elements (Blocks, parts)» updated with additional reliability information (ASD code, reliability figure, failure modes (including Failure Mode Id (FMI)) 6. In Functional modeller , Functional Architecture model is extended with dis-functional behaviour. 7. In the Functional modeller , Functional Architecture model (including dis-functional behaviour) is extended with fault detection aspects in behavioural view. 8. In the Functional modeller , Functional Architecture model (including dis-functional behaviour) is extended with Mission abort condition In the Functional modeller , single failure scenario simulation is performed and and single failure sequence diagrams are generated. 9. In the Functional modeller , single failure sequence diagrams are renamed with Failure Mode Id (FMI) 10. In the Functional modeller (safety profile), RBD models are developed. 11. In RM&T Analysis Tool, launch service «Get list of Functional Architectural model (IBD) simulation scenarios elements» 12. In RM&T Analysis Tool, launch service «Get RBD models from Safety Profile» 13. In RM&T Analysis Tool, FMEA table is completed with failure mode effects and detection from simulation scenario results 14. In RM&T Analysis Tool, RBD diagrams are analysed In Requirements manager tool, launch service «Get Reliability & Testability Performance System Result (MOE)»	list of MoE available verification report available (coverage report)
Artefacts provided as input of the activity 1. requirement list 2. functional model 3. physical model	**Artefacts** produced during the activity 1. System breakdown codification 2. Reliability allocation on basic elements 3. Functional model extended with dis-functional and fault detection behaviour 4. FMEA table 5. RBD	**Artefacts** which are the result of the activity Evidence on System Reliability and Testability results (requirements compliance)

Additional Comments: N/A

4.3 Change Impact Analysis

Engineering Method: Change Impact Analysis		
Purpose: the purpose is to support the analysis of the impact of a change of a CI on all correlated models, providing a traceability table/matrix		
Pre-Condition requirement list available functional model available physical model available reliability model available link traced Models are managed as Configuration Items (CI) Relationships between CI of System View and As Designed View are managed in PLM (a function is allocated to a system or logical equipment, a system or logical equipment implement a set of functions, a logical equipment is traced to a physical part (Equipment installation/assembly/part)	**Engineering Activity** 1. A Change of a requirement, a functionality or a CI arises 2. Find all impacted items in the tool chain through SEE environment (links traceability) 3. In PLM retrieve all CI that are impacted considering the different view 4. For each impacted domain specialist considers how the change impact the different objects in the models. 5. Discipline domain specialist evaluates if the main change drive a change in the domain model or not 6. Discipline domain specialist modifies local functional, physical or RM&T domain models according to change requirements	**Post-Condition** Traceability Table/Matrix available Navigation of TRACE relationships and visualization changes incorporated in the domain models.
Artefacts provided as input of the activity 1. requirement list 2. functional model 3. physical model 4. reliability model 5. traceability information in PLM	**Artefacts** produced during the activity 1. List of domain impacted Data 2. Traceability Matrix/Table 3. new functional, physical and RM&T model	**Artefacts** which are the result of the activity 1. Traceability Matrix/Table
Additional Comments: N/A		

4.4 Maintain Consistency between Multi-viewpoints Models

Engineering Method: Maintain Consistency between Multi-viewpoint Models		
Purpose: the aim of this method is to ensure that the different models in development during the preliminary phases of a project are consistent with each other. The information contained in the models shall not be incongruent. Any time a modification that impacts on other model will be made a notification of mismatch shall be sent to proper users. This Engineering Method is the natural follower of "Change impact analysis" EM.		
Pre-Condition requirement list available functional model available physical model available reliability model available (all models are considered configured managed in PLM)	**Engineering Activity** 1. change authorized in PLM 2. model change in a specific domain (Functional, Physical or RM&T) 3. Domain specialist send data update notification to all impacted models 4. impacts are evaluated and harmonized 5. change embodied in other models 6. manage and trace models baselines	**Post-Condition** 1. updated and consistent models available
Artefacts provided as input of the activity 1. requirement list 2. functional model 3. physical model 4. reliability model	**Artefacts** produced during the activity 1. authorized change 2. change notification 3. change impact report 4. models baselines	**Artefacts** which are the result of the activity 1. Updated models
Relevant to link between reliability models and functional / physical models, the steps 3. and 4. of Engineering Activity have to be considered in bidirectional way during the requirements analysis process, i.e. changes in functional / physical models (coming from analysis of functional / performance requirements) will generate update of reliability models and, vice versa, changes in reliability models (induced by analysis and verification of reliability requirements) will generate updated of functional / physical models (e.g. integration of new failed behaviour / scenarios, detection means, etc.)		

4.5 Provide Process Management

<table>
<tr><th colspan="3">Engineering Method: Provide Process Management</th></tr>
<tr><td colspan="3">Purpose: To provide the involved stakeholders, such as System engineer and Domain Experts, with “context-aware” support about the actions and the process part they have to apply within their daily activity. To gain access to the information about the system under development, including the monitoring of development status and process implementation. A possible instance intend to show how a support can be provided to those stakeholders that are involved in the change process.</td></tr>
<tr><td>Pre-Condition
Applicable process specification and Customized workflow are available.
This includes the involved stakeholders and boards (team members).
Project Area created in the ALM platform repository.

Instance related conditions:
Requirements list is available in PLM
System views are available in PLM:
- establish the traces among requirements, design models, verification cases and analysis reports</td><td>Engineering Activity
1. Configure the Workflow Manager platform through the desired “formal” process specification. (model transformation may be needed).
2. Assign roles in the Workslow Manager and notify them
3. Notify through the Workflow manager the stakeholders about their current task to be performed.
4. On the work bench, provide relevant (context aware) task details to the stakeholders (on demand)
5. Display updated list of task related events
6. Display available task progress information (activity staus changes) in the tracking task-board.
7. Stakeholder asks for system information and links to development data through proper queries issued through the dedicated interface.
8. Workflow management retrieves system information and traces from ALM platform through the established links (OSLC linked data)
9. Workflow management evaluates progress information
10. Task relevant (updated) information are displayed to stakeholders through dedicated work item perspectives of the work bench.</td><td>Post-Condition
1. All the needed stakeholders have been contacted and informed about their current duties
2. Each stakeholder got exhaustive information about the task to be performed.
3. Each stakeholder has the need awareness of the assigned task according to its role
4. Monitoring information about the on-going process are available to the stakeholders
5. Link to Integrated views about relevant system aspects is made available.
6. Task related events are notified according to planned stakeholder’s need to know.</td></tr>
<tr><td>Artefacts provided as input of the activity
1. Relevant process specification formalized (e.g SPEM)
2. Relevant workflow (collaboration process) formalized through BPMN2 or equivalent.
Instance related data:
3. Requirements list and traces to models and V&V
4. System Configuration Information.</td><td>Artefacts produced during the activity
1. Task related Work Items
2. Process monitoring data
3. Requests for system information
4. Required system view for display
5. Task related events</td><td>Artefacts which are the result of the activity
1. Task relevant messages to the stakeholders
2. Log about performed activities
3. Updated progress data
4. Updated System configuration information (if applicable)</td></tr>
<tr><td colspan="3">Additional Comments
Workflow Management services shall be configured through well recognized process specification standards. Currently envisaged standards are SPEM (specification) and possibly BPMN2 (enactment).</td></tr>
</table>

4.6 Provide configuration Control

Engineering Method: Provide Configuration Control		
Purpose: provide configuration control not only of the physical views but also of requirement and functional views in order to manage and reuse these artifacts for similar product/capability classes (related to the Baseline management and Release Process Activity)		
Pre-Condition Functions are managed as Activities of Activity Diagrams in functional modeler (SysML modeling). Systems/Sub-Systems/Logical Equipment are managed by Blocks in MBSE tool (SysML modeling). As-Designed view available Systems/Subsystems/Logical Equipment/ Functions are managed as Configuration Item in PLM Tool. A Functional Specification defined in a SysML Model has been frozen as Baseline at the end of Functional Analysis (Black box activity diagram) applicable to a specific configuration A Functional Specification defined in a SysML Model has been frozen as Baseline at the end of Design Synthesis (White box activity diagrams) applicable to a specific configuration	**Engineering Activity** 1. In PLM select the SYSTEM under analysis 2. In PLM ask for List of System Functionalities 3. Request is forwarded by SEE environment to functional modeller 4. In the SEE environment, the List of all functions is assembled and sent back to the PLM tool 5. In PLM receive functions 6. In PLM the developer associates information related to applicability to the imported Functions 7. In PLM correlate System View CI to As-Designed View CI 8. In PLM , select the SYSTEM under analysis 9. In PLM , launch service “Get List of All Sub-system Functionalities” 10. Request is forwarded by SEE environment to Functional Modeller 11. In the SEE Environment, for each SUBSYSTEM the List of allocated functions is assembled and send back to PLM tool 12. In PLM , receive SUBSYSTEM functions 13. In PLM, the developer introduces associates information related to applicability to the imported functions	**Post-Condition** Systems View Management in PLM tool with Applicability management of Functionalities defined in MBSE tool (SysML modelling). Management of Commonalities and Comparison of different Functional Configuration in PLM. Management of traceability from System View CI to As-Designed View CI (e.g. Function to Part Number) in the PLM tool.
Artefacts provided as input of the activity 1. Activity Diagrams (SysML) 2. Interface Identification Diagram (IBD, SysML) 3. Functional Specification Baseline	**Artefacts** produced during of the activity 1. System Functionalities List 2. Sub-System Functionalities List	**Artefacts** which are the result of the activity 1. System View 2. System View links to other views
Additional Comments: N/A		

4.7 Heterogeneous Simulation

Engineering Method: Heterogeneous Simulation		
Purpose: the aim of this method is to assess system concepts through simulation:		
Pre-Condition Requirements list available Physical models available Functional models available (all models are considered configured and executable)	**Engineering Activity** 1. The analyst engineer requests the list of the available system requirements 2. The analyst engineer request the list of the available physical models related to Fuel System 3. The analyst engineer request the list of the available Functional models related to Fuel System 4. The analyst engineer analyses the architectural and functional models, and select the allocated set of requirements that impact the model 5. The analyst engineer launches the Simulation tool and analyses components and system performances 6. The analyst engineer verifies the requirements against the calculated performances 7. The analyst engineer set up the adjustement to the models and provide the new releases 8. The analyst engineer requests the list of the updated physical model 9. The analyst engineer can run combined simulation using two different simulation models	**Post-Condition** 1. verified requirements by simulation of the fuel system Simulated Fuel System
Artefacts provided as input of the activity 1. requirements list 2. functional models 3. physical models	**Artefacts** produced during the activity 1. Fuel System model	**Artefacts** which are the result of the activity 1. Updated models
Additional Comments		

5 Terms, Abbreviations and Definitions

ALM	Application Life-Cycle Manager
BPMN	Business Process Model and Notation
CI	Configuration Item
EIMSS	Enhanced Integrated Monitoring and Support System
GSS	Ground Support System
MBSE	Model Based Systems Engineering
MC	Maintenance Computer
MoE	Means of Evidence
OSLC	Open Services for Lifecycle Collaboration
PLM	Product Life Cycle Manager
RBD	Reliability Block Diagram
RM&T	Reliability, Maintainability & Testability
SE	Systems Engineering
SEE	System Engineering Environment
SPEM	Software & Systems Process Engineering Meta-model
SysML	Systems Modelling Language
V&V	Verification & Validation

Table 11: Terms, Abbreviations and Definitions

6 References

CRYSTAL consortium; 2013	Annex I - "Description of Work"
CRYSTAL consortium; 2013	D202.010 “Use case Description”