|  |
| --- |
| UNCLASSIFIED |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | REFERENCES | ISSUE | DATE |
| logoIRT-StExupery |  |  |  |  |
|  |  |  |  |  |
| logoAirbus |  |  |  |  |
|  |  |  |  |  |
| logoONERA |  |  |  |  |
|  |  |  |  |  |
| logoRR |  |  |  |  |
|  |  |  |  |  |
| DASAVIA |  |  |  |  |
| logoSAFRAN |  |  |  |  |
| logoThales |  |  |  |  |

|  |
| --- |
|  |
| Streamlining Development Assurance  µXAV Process Definition |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Page : 1 / 46 |

|  |
| --- |
| This document is the intellectual property of RESSAC members It may not be used, reproduced, modified or disclosed without its authorization.  RESSAC members Proprietary Data. |

signatures

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | REFERENCES | APPROVERS |
|  |  | Fonction 1 |
|  |  |  |
|  |  | Name 1 |
|  |  | Fonction 2 |
|  |  |  |
|  |  | Name 2 |
|  |  | Fonction 3 |
|  |  |  |
|  |  | Name 3 |
|  |  | Fonction 4 |
| logoRR |  |  |
|  |  | Name 4 |
|  |  | Fonction 5 |
| DASAVIA |  |  |
|  |  | Name 5 |

Table of contentS

1. Document issues 5

2. PURPOSE AND SCOPE OF THE DOCUMENT 6

3. ORGANIZATION 6

4. LIFE CYCLE PROCESSES 6

4.1 Principles 6

4.2 Description 7

5. Process Activities 8

5.1 Planning process 8

5.1.1 µXAV process definition 8

5.1.2 Collective review (µXAV process definition verification) 8

5.2 Increment definition 9

5.2.1 Increment definition 9

5.3 µXAV level 10

5.3.1 µXAV specification 10

5.3.2 µXAV functional decomposition 11

5.3.3 µXAV specification validation 11

5.3.4 µXAV tests 11

5.4 System Level 12

5.4.1 µXAV architecture 12

5.4.2 System specification 13

5.4.3 System specification verification 13

5.4.4 µXAV architecture verification? 14

5.4.5 Inter-System specification verification 14

5.4.6 System integration and tests 15

5.4.7 Inter-System tests? 15

5.5 Software items 15

5.5.1 SW item type 1: SCADE 15

5.5.2 SW item type 2: SPARK 17

5.5.3 SW item type 3: C and formal method 20

5.6 Hardware items 20

5.6.1 HW item type 1: method name 20

5.6.2 HW item type 2: method name 20

5.7 Equipment items 20

5.7.1 Equipment item type 1: method name 20

5.7.2 Equipment item type 2: method name 20

6. Environments 20

6.1 Tools 20

7. CONFIGURATION MANAGEMENT PROCESS 21

7.1 Configuration Management Environments 21

7.2 Change management principles 21

7.3 Life Cycle Data 21

8. Appendixes 22

8.1 Mission scenario template 22

# Document issues

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue** | **Author(s)** | **Updating purpose** |
| 24/10/2016 | First drafts | F.Pothon | Creation of the document based several meeting discussions with all RESSAC Partners |

# PURPOSE AND SCOPE OF THE DOCUMENT

The purpose of this plan is to describe the µXAV life cycle, developed in the scope of the RESSAC project, the development processes and all integral processes.

RESSAC project will define overarching properties and criteria for development assurance. In order to avoid any assumptions on the satisfaction of these IPs and criteria, the processes and data defined for the use case are identified independently of the definitions of overarching properties and criteria. Matching assessment between use case processes and RESSAC criteria will be performed in parallel in order to consolidate these criteria.

# ORGANIZATION

In the scope of RESSAC project, the activities are shared between the project partners.

# LIFE CYCLE PROCESSES

## Principles

The product life cycle is incremental, consisting in developing progressively a limited number of new functions/features. This incremental approach creates an iterative life cycle. Iteration consists in adding/modifying and verifying some features in the data developed during the previous iteration cycles.

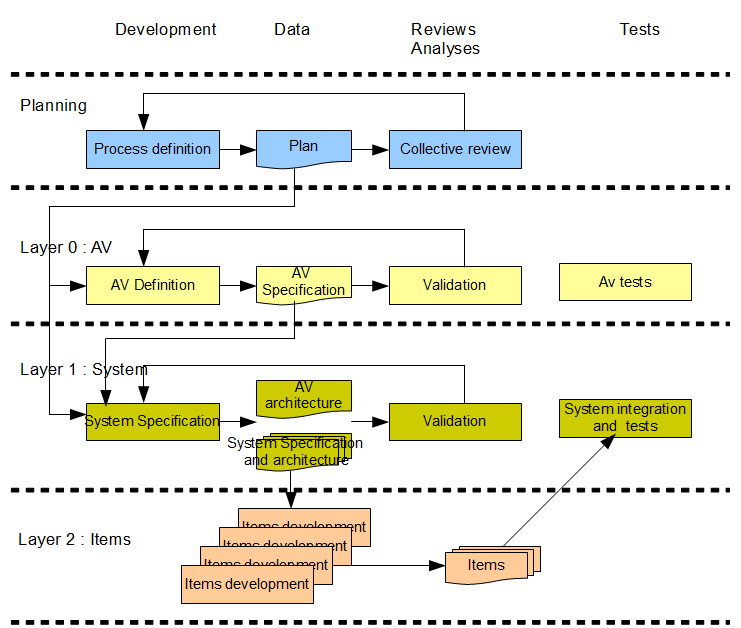
The complete development of the µXAV includes several abstraction layers:

* Layer 1: µXAV level. It consists in defining the µXAV specification and the µXAV external interfaces
* Layer 2: System level. This layer starts with the definition of the “system”, the specification of each system and the architecture of each system. This architecture includes the definition of each items of the system and their interfaces
* Layer 3: It is the item level. The development of each item is separately developed based on the system specification allocated to this item. Each item may use different methods and level of refinements (internal development tiers). Items are defined in the system architecture, and may be
  + A pure software items. That’s mean that the item is developed independently of the target. Integration HW/SW is performed at system level. The item is delivered for system integration in a form that will be compatible with the system integration
  + One or several hardware components, complex or not. An electronic card is considered as a hardware item
  + An “equipment” where a software is integrated in the hardware computer.

The incremental approach may be applied at the 3 abstraction layers.

## Description

The following figure displays the activities to be performed on the 3 layers



# Process Activities

## Planning process

This process consists in identifying the applicable life cycle and the activities to be performed. A first release of the µXAV process definition is provided at the beginning of the project. Then this definition may be updated iteratively as necessary. The µXAV process definition is accepted through a collective review by the RESSAC project partners.

### µXAV process definition

* Description:

The purpose of this activity is to identify the activities, environment, methods and responsibilities for the development, verification/validation of the µXAV.

* Methods:

The process definition is a textual document. It is developed through a set of collective workshops and discussion with various partners and experts. Each collective workshop are prepared through several exchanges and followed by a writing activity by data author.

* Environment:

No particular environment is necessary for this activity, only textual editor.

* Responsibilities: TBD
* Inputs: No specific inputs
* Outputs

Process Definition document

### Collective review (µXAV process definition verification)

* Description:

The purpose of this activity is to verify the applicability and correctness of the process definition.

* Methods:

This activity consists in a review performed during a face to face meeting. All attendees read collectively the data under verification, and update it. Consensus is required to implement the changes.

The history section of the data records the date of the review, the main changes, and attendees

In case of remaining topics needing further investigations, discussions TBD

* Environment:

No particular environment is necessary for this activity, only textual editor.

* Responsibilities: TBD
* Inputs: No specific inputs
* Outputs

TBD

## Increment definition

As stated in [§4.1 Principles](#_Principles), the µXAV is incremental. So the first increment includes a subset only of required functions. After this first increment identification of new functions are identified. These functions can be added at any layer.

### Increment definition

* Description:

The purpose of this activity is to identify the new functions to be added at µXAVand/or system layer and/or items level. Along with this identification, an impact analysis is conducted, identifying the change impact and the need for re-verification.

* Methods:

The increment definition is formalized in an Increment Definition File, that is a textual document. It is developed during a collective workshop.

* Transition criteria : End of last increment
* Environment:

No particular environment is necessary for this activity, only textual editor.

* Responsibilities: TBD
* Inputs

Data produced during last increment

* Outputs

Increment Definition File

## µXAV level

Three activities are performed at µXAV level: The µXAV specification, the µXAV specification validation and the µXAV tests.

### µXAV specification

* Description:

The µXAV specification is mainly based on mission scenario description. The Mission Scenarios address the different operational modes and the possible degraded modes in case of failure, or abnormal environment conditions.

These scenarios are supplemented as necessary with additional requirements and constraints such as performances aspects. These additional requirements and constraints do not duplicate the scenarios but express characteristics, conditions that cannot be included in any scenarios.

The foreseeable conditions in which the µXAV will operate are identified. These conditions define the normal and abnormal inputs and conditions.

In parallel of mission scenario development, the µXAV external interfaces are defined.

Iterations between the different element of the µXAV specification (Mission Scenarios, additional requirements, external interfaces and foreseeable conditions) are performed to ensure the consistency of the complete specification.

* Methods:

The µXAV specification is a textual document.

Each scenario identifies step by step the conditions and inputs and observable properties. The combination of all mission scenarios should be representative of all operating conditions.

A scenario description uses the template defines in appendix A of the document

This specification is used as input for each system development. The Mission Scenarios may be directly used as inter-system integration verification cases, while additional requirements will be the purpose of additional validation activities.

* Transition criteria ?
* Environment:

No particular environment is necessary for this activity, only textual editor.

* Responsibilities: TBD
* Inputs

The µXAV specification is developed based on the knowledge and background of the RESSAC project patterns of the desired behaviour and of the foreseeable operating conditions.

* Outputs

µXAV specification

### µXAV functional decomposition

To be confirmed

### µXAV specification validation

* Description:

The µXAV specification is validated by the different stakeholder having the knowledge of the desired system behavior and foreseeable operating conditions.

* Methods:

The validation is performed through a proof-reading of the document.

* Transition criteria ?
* Environment:

No particular environment is necessary for this activity

* Responsibilities: TBD
* Inputs

µXAV specification

* Outputs

TBD

### µXAV tests

* Description:

This activity consists in verifying the compliance of the µXAV to its specification.

* Methods:

After integration of the systems, this activity consists in piloting the µXAV, applying the mission scenarios of the µXAV specification.

Additional mission scenarios may be developed to checks further operating conditions combination as necessary, or performances.

* Transition criteria ?
* Environment: Integrated µXAV
* Responsibilities: TBD
* Inputs

µXAV specification

Integrated µXAV

* Outputs

TBD

## System Level

The following activities are performed at system level: µXAV architecture, and for each system, system specification, the system specification validation and the system integration and tests.

### µXAV architecture

* Description:

The µXAV architecture is developed identifying the several systems and their interfaces.

* Methods:

TBD.

* Transition criteria ?
* Environment:

TBD

* Responsibilities: TBD
* Inputs

The µXAV specification

Outputs

µXAV architecture

### System specification

* Description:

The activity may be conducted separately on each system. It consists in defining the expected behavior of the system. Then for each system the architecture is defined, identifying the several items, and their interfaces.

Iterations between the different element of the system specification (system requirements and architecture) are performed to ensure the consistency of the complete specification.

* Methods:

Modelica is used for system specification

System architecture: SysML

* Transition criteria ?
* Environment:

SysML

Modelica

* Responsibilities: TBD
* Inputs

µXAV specification

µXAV architecture

* Outputs

System specification (Modelica + SysML)

### System specification verification

* Description:

Each system specification is verified for correctness and consistency.

* Methods:

The verification is performed through modelica simulation. (To be completed)

* Transition criteria ?
* Environment:

Modelica simulation

* Responsibilities: TBD
* Inputs

System specification (Modelica model)

* Outputs

TBD

### µXAV architecture verification?

* Description:

A verification of consistency of the decomposition of the system into items is performed

### Inter-System specification verification

* Description:

A verification of consistency of all system specification is performed. This verification is based on the µXAV specification and µXAV architecture. Its purpose is to detect the complete implementation of the mission scenario and compliance to the additional requirements. System requirements that do not participate to any µXAV specification items are identified and justified.

* Methods:

TBD

* Transition criteria ?
* Environment:

TBD

* Responsibilities: TBD
* Inputs

µXAV specification

µXAV architecture

All System specification

* Outputs

TBD

### System integration and tests

* Description:

This activity consists in verifying the compliance of each system to its specification. This activity is performed separately on each system

* Methods:

This verification is based on the use of modelica using an incremental approach. When an item is available, it is plugged into the models and replaced its specification. Then the simulation performed for system specification verification is re-run and compared to the simulation results.

This activity is re-entered each time new items(s) are available.

* Transition criteria ?
* Environment: Integrated system
* Responsibilities: TBD
* Inputs

System specification

Items

* Outputs

TBD

### Inter-System tests?

* Description:

## Software items

Three technologies are identified for software items development (SCADE model, SPARK, and C language+formal method)

### SW item type 1: SCADE

#### Methods definition

* Description:

The purpose of this activity is to define guidelines to apply the chosen methods on the item.

* Methods:

TBD

* Environment:

No particular environment is necessary for this activity, only textual editor.

* Responsibilities: TBD
* Inputs:

No specific inputs

* Outputs

SCADE guidelines

#### Item development

* Description:

The purpose of this activity is to develop the SCADE model to implement the system requirements allocated to the item

* Methods:

SCADE model

* Environment:

SCADE editor

* Responsibilities: TBD
* Inputs:

System specification allocated to the item

* Outputs

SCADE model

#### Item verification

* Description:

The purpose of this activity is to verify the compliance of the SCADE model to the system specification allocated to the item

* Methods:

Model simulation

* Environment:

SCADE simulation

* Responsibilities: TBD
* Inputs:

System specification

SCADE model

* Outputs

Simulation procedures and results

### SW item type 2: SPARK

#### Methods definition

* Description:

The purpose of this activity is to define guidelines to apply the chosen methods on the item.

* Methods:

TBD

* Environment:

No particular environment is necessary for this activity, only textual editor.

* Responsibilities: TBD
* Inputs:

No specific inputs

* Outputs

SPARK guidelines (requirements and code)

Formal analysis description

#### Item requirement development

* Description:

The purpose of this activity is to develop the item requirements (in SPARK) to implement the system requirements allocated to the item

* Methods:

SPARK

* Environment:

TBD

* Responsibilities: TBD
* Inputs:

System specification allocated to the item

* Outputs

Item requirements

#### Item requirement verification

* Description:

The purpose of this activity is to verify the compliance of the item requirements to the system specification allocated to the item

* Methods:

Review?

* Environment:

TBD

* Responsibilities: TBD
* Inputs:

System specification

Item requirements

* Outputs

TBD

#### Item implementation

* Description:

The purpose of this activity is to implement the item requirements into source code, and to produce the executable object code

* Methods:

Ada

* Environment:

Gnat Ada environment

* Responsibilities: TBD
* Inputs:

Items requirements

* Outputs

Source code

Executable object code

#### Item verification

* Description:

The purpose of this activity is to verify the compliance of the item implementation to the item requirements

* Methods:

The activity consists in running GNATProve. The tool detects all cases where the contracts are not satisfied.

* Environment:

GnatProve

* Responsibilities: TBD
* Inputs:

Item requirements

Source code

* Outputs

Analysis report

### SW item type 3: C and formal method

## Hardware items

### HW item type 1: method name

### HW item type 2: method name

## Equipment items

### Equipment item type 1: method name

### Equipment item type 2: method name

# Environments

## Tools

# CONFIGURATION MANAGEMENT PROCESS

## Configuration Management Environments

Github

## Change management principles

## Life Cycle Data

# Appendixes

## Mission scenario template