|  |
| --- |
| 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 specification validation 11

5.3.3 µXAV functional decomposition ? 11

5.3.4 µXAV architecture 12

5.3.5 µXAV architecture verification? 12

5.3.6 µXAV integration and verification 13

5.4 System Level 14

5.4.1 System specification and architecture 14

5.4.2 System specification validation 15

5.4.3 Inter-System specification verification 15

5.4.4 System integration (implementation) and verification 16

5.4.5 Inter-System verification 16

5.5 Software items 17

5.5.1 SW item type 1: SCADE 17

5.5.2 SW item type 2: SPARK 18

5.5.3 SW item type 3: C and formal method 21

5.6 Hardware items 21

5.6.1 HW item type 1: method name 21

5.6.2 HW item type 2: method name 21

5.7 Equipment items 21

5.7.1 Equipment item type 1: method name 21

5.7.2 Equipment item type 2: method name 21

6. Environments 21

6.1 Tools 21

7. CONFIGURATION MANAGEMENT PROCESS 22

7.1 Configuration Management Environments 22

7.2 Change management principles 22

7.3 Life Cycle Data 22

8. Appendixes 23

8.1 Mission scenario template 23

# Document issues

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue** | **Author(s)** | **Updating purpose** |
| 16/12/2016 | First drafts | F.Pothon | Creation of the document based on several meeting, discussions, and contributions fromRESSAC 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 OPs 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.

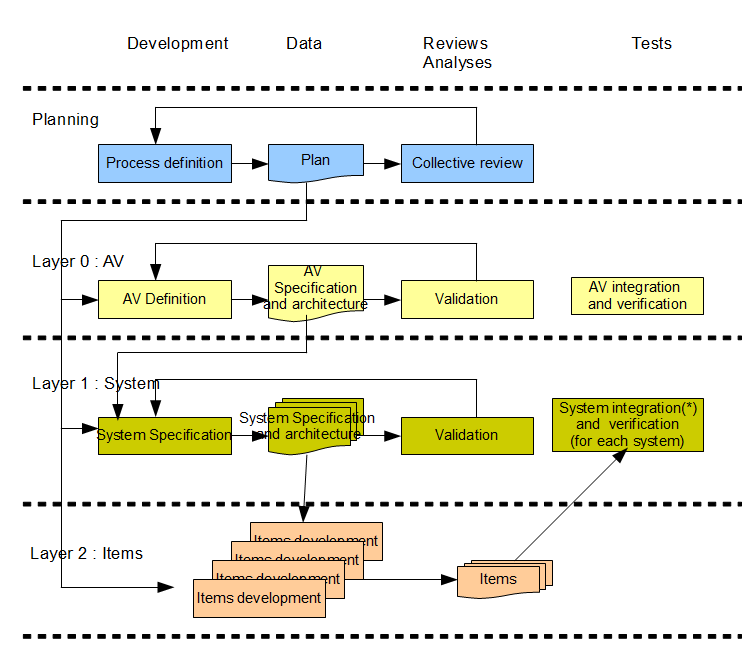
The complete development of the µXAV includes several abstraction layers:

* 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



(\*) System integration is the integration of items to implement the system

# 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
  1. 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:
  1. Updated Process Definition document

## Increment definition

As stated in [§4.1 Principles](#_Principles), the development of µXAV is incremental. So the first increment includes a subset only of desired bahavior. 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 µXAV and/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 : The minimum criteria to launch the increment definition are
  1. End of last increment
* Environment:

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

* Responsibilities: TBD
* Inputs
  1. Data produced during last increment
* Outputs
  1. Increment Definition File

## µXAV level

Three activities are performed at µXAV level: The µXAV specification, the µXAV specification validation and the µXAV integration (implementation) verification.

### 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: The minimum criteria to launch the µXAV specification are:
  1. Applicable sections of Process definition document agreed
  2. Increment Definition File identified
* 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
  1. µXAV specification

### 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: The minimum criteria to launch the µXAV specification validation are:
  1. Applicable sections of Process definition document agreed
  2. µXAV specification under identified
* Environment:

No particular environment is necessary for this activity

* Responsibilities: TBD
* Inputs
  1. µXAV specification
* Outputs
  1. TBD

### µXAV functional decomposition ?

To be confirmed: Part of µXAV specification or another artefact?

* Description:
* Methods:
* Transition criteria
* Environment:
* Responsibilities: TBD
* Inputs
* Outputs

### µXAV architecture

* Description:

The µXAV architecture is developed identifying the several systems and their interfaces. Architectural mitigation introduced by safety analysis are addressed

* Methods:

SysML

* Transition criteria
* Environment:

TBD

* Responsibilities: TBD
* Inputs
  1. µXAV specification
  2. Architectural mitigation provided by system analysis
  3. µXAV functional decomposition
* Outputs
  1. µXAV architecture

### µXAV architecture verification?

* Description:

A verification of consistency of the decomposition of the µXAV into systemsinto items is performed with regards to the µXAV specification and functional decomposition

* Methods:
* Transition criteria
* Environment:
* Responsibilities: TBD
* Inputs
  1. µXAV specification
  2. Architectural mitigation provided by system analysis
  3. µXAV functional decomposition
* Outputs
  1. TBD

### µXAV integration and verification

* Description:

This activity consists in integrating the systems progressively and 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 for each system. It includes system specification, the system specification validation and the system integration (implementation) and and related verification.

### System specification and architecture

* 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 ? µXAV specification exists and upper level functional decomposition is available with interfaces definition.
* Environment:

SysML

Modelica

* Responsibilities: TBD
* Inputs

µXAV specification

µXAV architecture

* Outputs

System specification (Modelica + SysML)

### System specification validation

* 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

### 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 (implementation) and verification

* 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 verification

* Description:
* Methods:
* Transition criteria ?
* Environment: Integrated system
* Responsibilities: TBD
* Inputs
* Outputs

TBD

## 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