CENTRE NATIONAL D'ÉTUDES SPATIALES

# GAIA – CU6

## Presentation of Product Assurance and Engineering Dispositions – PAED

**Prepared by CNES**

**2nd Workshop, Brussels 12-13 October 2006**

CENTRE NATIONAL D'ÉTUDES SPATIALES

## PAED – GENERALITIES

## A set of dispositions for software development and maintenance

- **The PAED describes the approach (for PA and engineering domains) to be taken into account for software development and maintenance**

- **This Standard provides dispositions, rules, methods and procedures that must be followed by the software project organization, software engineering, software product assurance and software configuration management.**



## PAED Objectives

- **To be applicable to all CUs : to speak the same "language"**
  - Same terminology (based on ECSS)
- **To obtain a satisfactory software quality level**
  - Reliable
  - Easy to maintain
  - With Performance and load capacity
- **To save time solving common problems**
  - Same project organization and development cycles (Iterative, 6 month cycle)
  - Same deliverables and basic engineering dispositions
  - Same tools for all CUs
- **Essential for a project**
  - **Where the final system will integrate components from**
    - Hundreds of developers from a large number of institutes
  - **Where corrective and enhancements maintenance will cover the period 2012-19**

## Scope of PAED

- ORGANIZATION
- SOFTWARE DEVELOPMENT
  - DEVELOPMENT LIFE CYCLE
  - DOCUMENT DELIVERABLES
  - SPECIFICATION
  - DESIGN
  - CODING
  - UNIT AND SOFTWARE INTEGRATION TESTS
  - VALIDATION
  - DELIVERY AND ACCEPTANCE
  - SYSTEM INTEGRATION
  - PERFORMANCE AND MARGIN MANAGEMENT
  - SOFTWARE MAINTENANCE

**Plan of the PAED document**



## Scope of "Product Assurance and Engineering dispositions for scientific development

- QUALITY ASSURANCE PROCESSES
  - VERSIONING, MODIFICATION AND CONFIGURATION MANAGEMENT
  - NON CONFORMANCE MANAGEMENT
  - PLANNING AND RESOURCE MANAGEMENT
  - ACTION MANAGEMENT

## Some basis of a software development

- DOCUMENTATION
  - A software product can not be produced without at least describing :
    - **The functions of the software**
    - **The non functional characteristics of the application (performance, etc.)**
    - **The interfaces (internal and external)**
    - **The design of the software**
    - **The software tests**
  - A software product can not be used without at least at least providing the evidence that :
    - **All requirements have been fulfilled**
- REQUIREMENTS MANAGEMENT
  - Requirements are the basis of many development activities so they shall be clearly identified



## Identification of requirements in the PAED document

- **Requirements in all GAIA documents shall be compliant with this identification convention**

**OUx** : organization unit ID (CU1, CU2, etc.) or DPC (CNES, ESAC, etc.)
**Source** is: WP number or a non ambiguous acronym
**Type** is: T: Technical; S: Scientific; Q: Quality, M: Management
**Scope** is: SPEC / DESG / COD / VALD / ACPT / MAIN / OPS / FUNC / PLAN / SUPP / PERF / RISK / CTRL
**ID** : a number used to differentiate requirements of a same scope, type, source and CU.

Draft / Approved / Obsolete

Auto. / Man.

| *Ref: CUx-SOURCE -TYPE-SCOPE-ID* | Version: C.v | Category | Priority | Verification | Status |
|---|---|---|---|---|---|
| Description of the requirement | | | | | |

HIGH / MEDIUM / LOW

**C**: Identifier of the development cycle number
**v**: Version number of the requirement

Category (list of target for transverse and science functions to be defined)

*Explanatory text in Italics – not part of the requi*

## Identification of dispositions, rules, methods and procedures in the PAED document – example

**Full ID :** CU1-WP103-M-PLAN-61
**Short ID :** M-PLAN-61

Note : CUx-WPXXX could be omitted in order to have short identifiers within a document. But the full identifier is used to refer any external requirements (defined outside the document)

| | | | | | |
|---|---|---|---|---|---|
| [CU1-WP103-]M-PLAN-61 | 1.1 | | HIGH | MAN | Approved |
| | The cycles are described in the development plan (SDP) according to the following criteria:<br>–Objectives,<br>–Activities planned, completion criteria. (e.g. acceptance, internal review)<br>Dates of reviews should be in accordance with the DPAC schedule and phases (see [**AD2**]) | | | | |

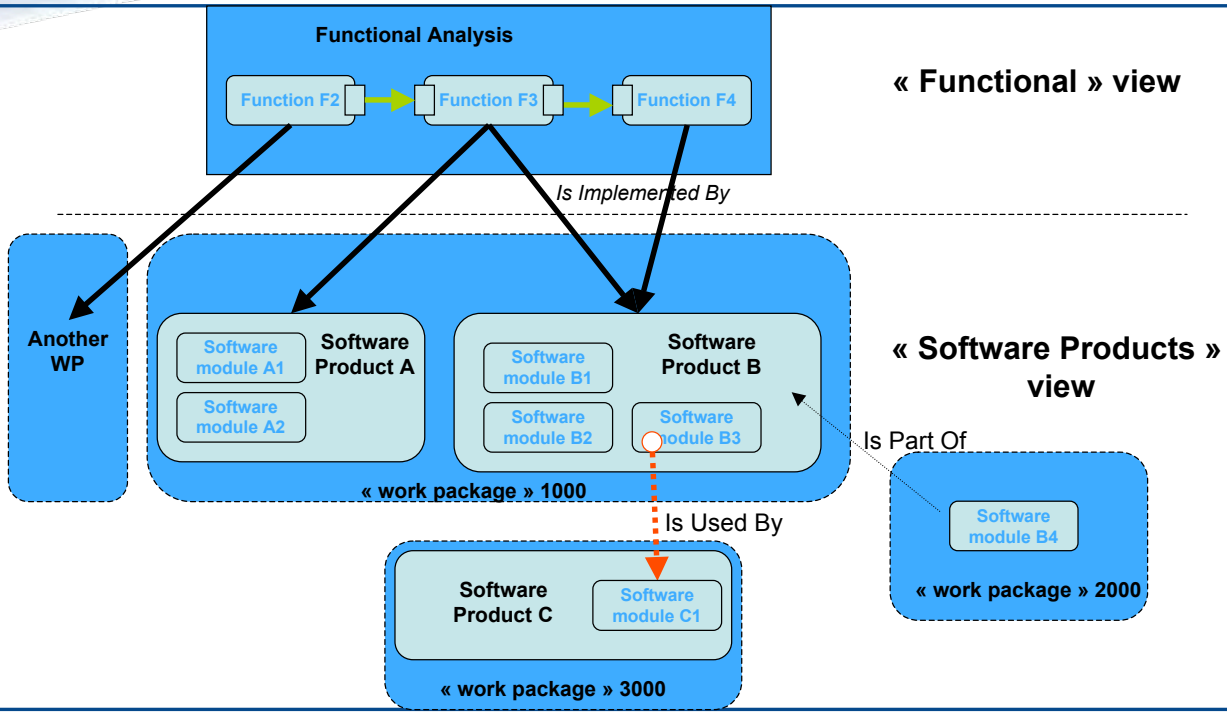


## A common terminology (ECSS based)

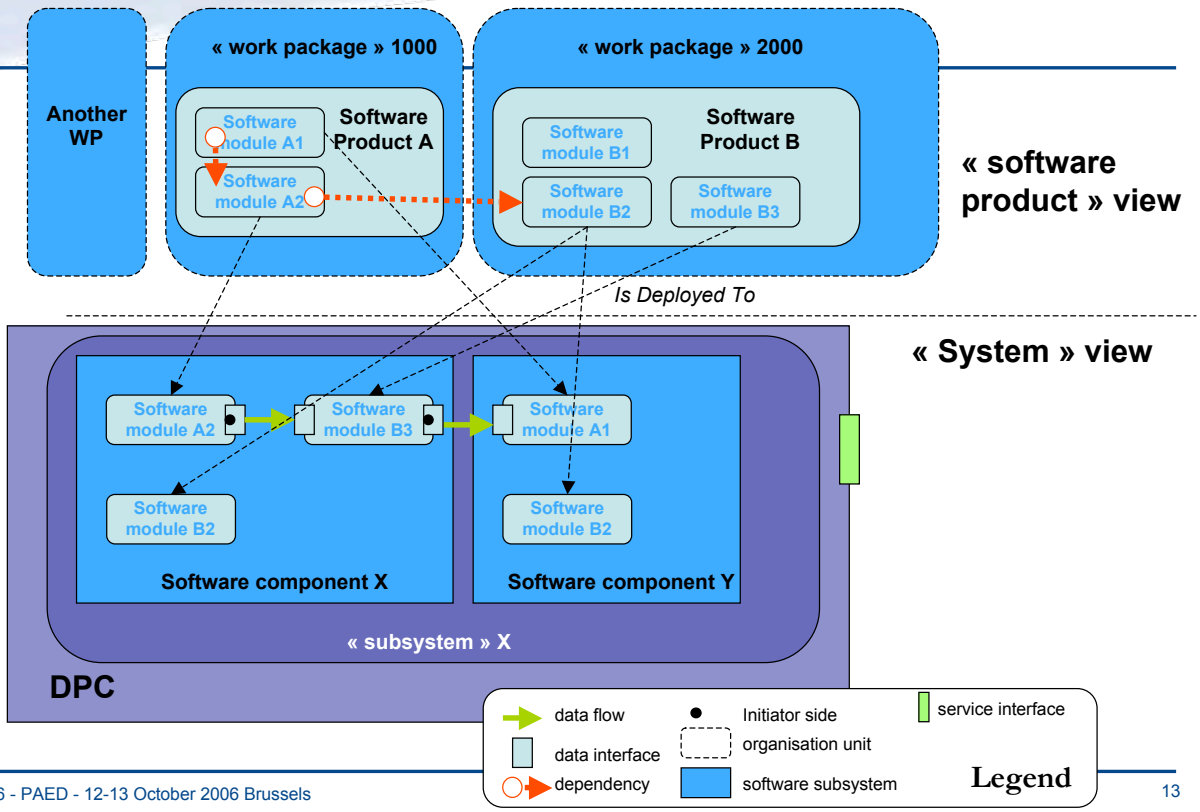
- acceptance / system integration / software validation / tests
- interface
- function
- scientific algorithm
- software component / software product / software module
- system / subsystem
- non conformance minor / major

# ORGANISATION : PRODUCED SOFTWARE

## Mapping functions to software products

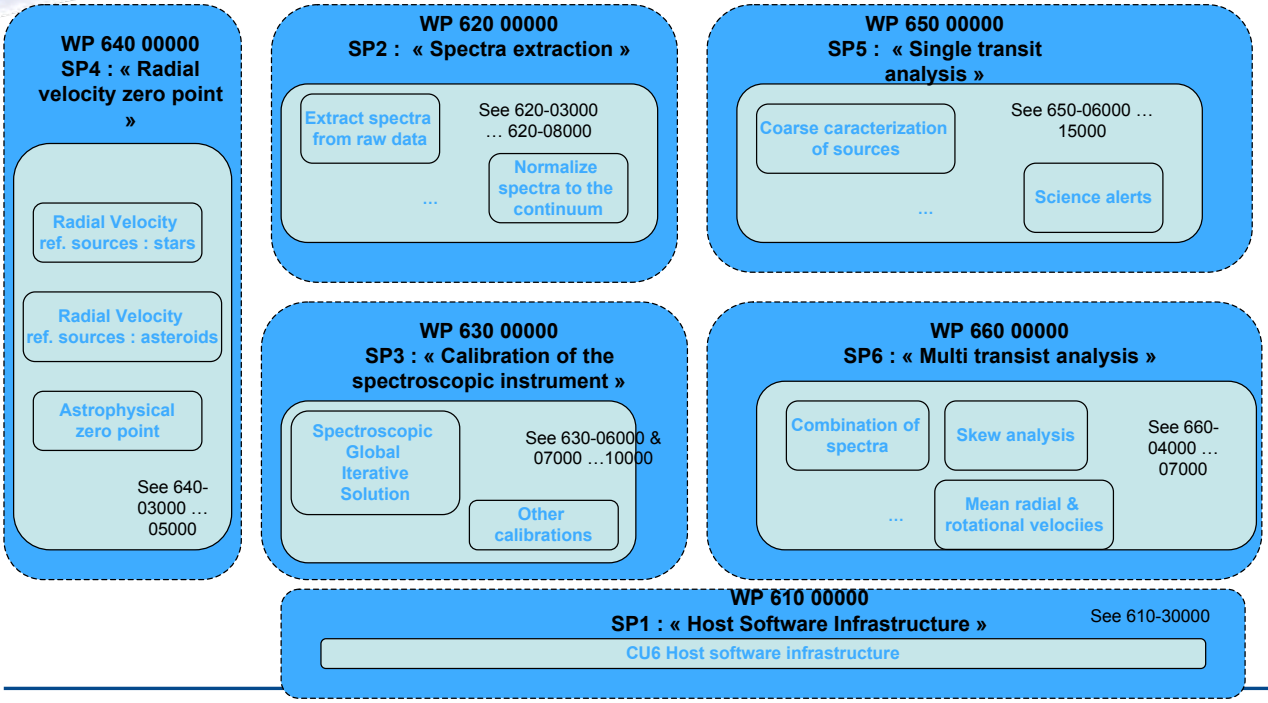


# Mapping software products to system





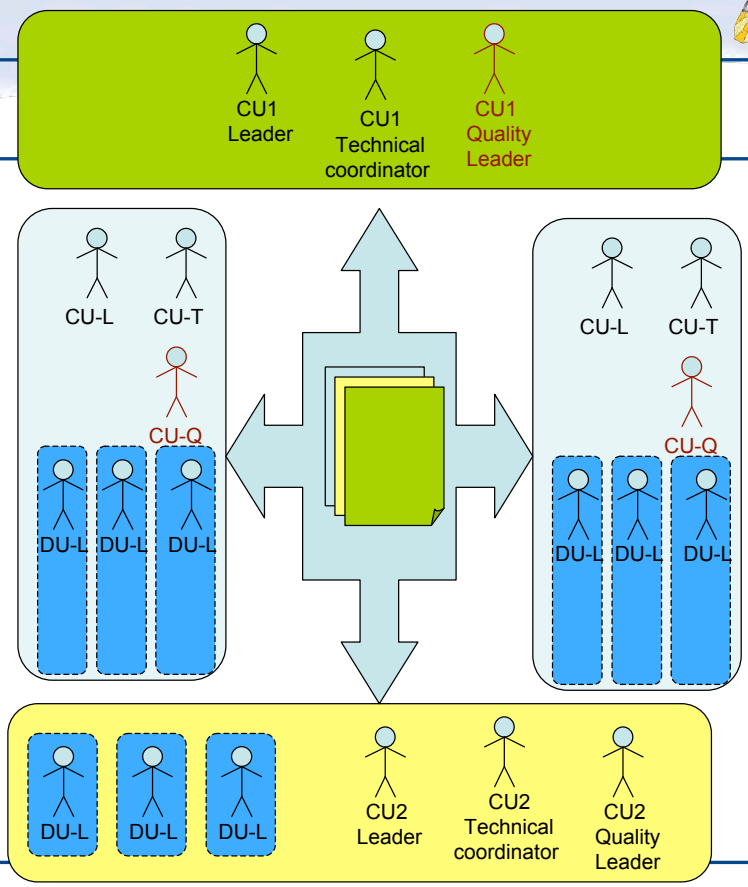

# CU6 : software products view



# ORGANISATION : ACTORS & ROLES

## Roles and organisation

- **For each CU:**
  - **CU-L David Katz** (Leader / Scientific coordinator)
  - **CU-T Anne J-A Piccolo / Nathalie Gerbier** (Technical coordinator)
  - **CU-Q Anne Thérèse Nguyen** (Quality Assurance Leader)
  - **DU/WP Leaders**
    **Yves Viala**
    **Mark Cropper**
    **Gérard Jasniewicz**
  - **Developer team**

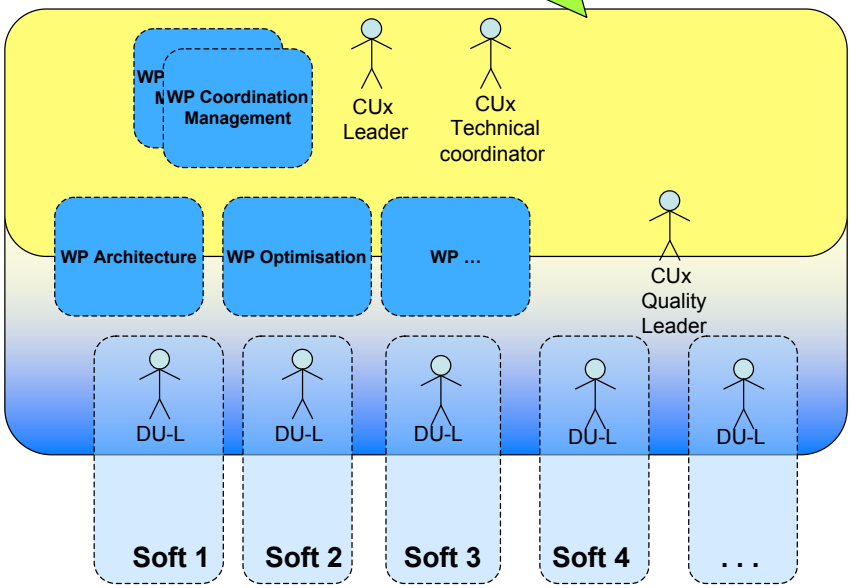


## Activities and process

■ CU : Different levels of decision

-Decisions should be taken at the « right » level to be efficient, to avoid blocking or cyclic problems.
- Decisions should be traced and applied.

Management and Technical board
=
CU6 Steering Committee

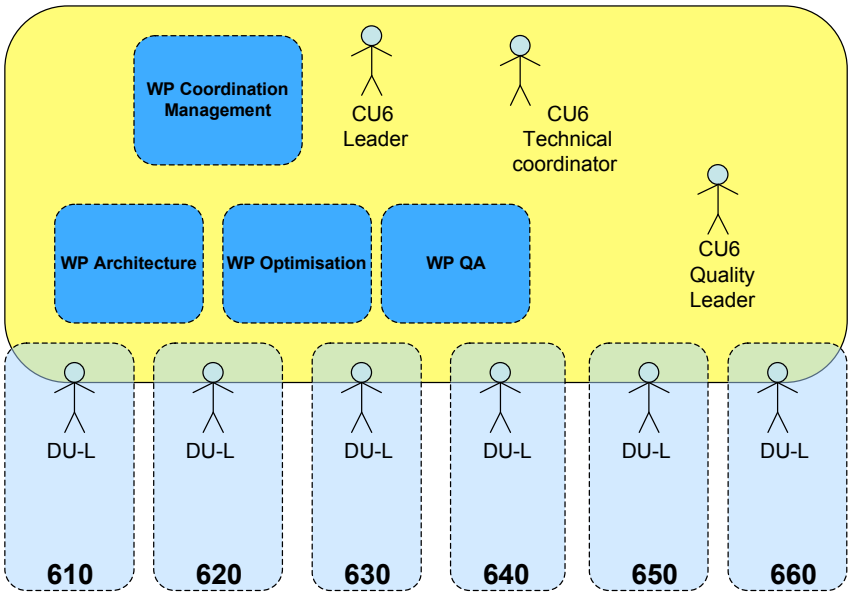



WP decision level



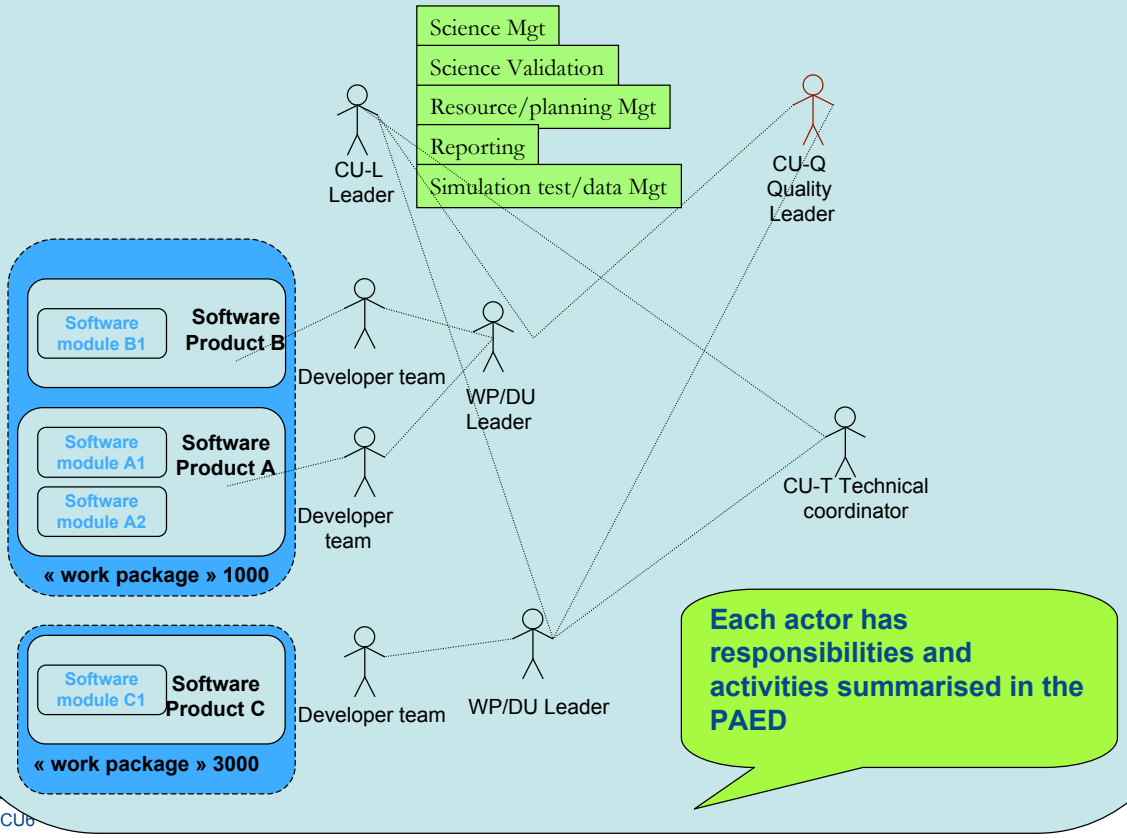
## CU6 : actors and roles

CU6 Steering Committee



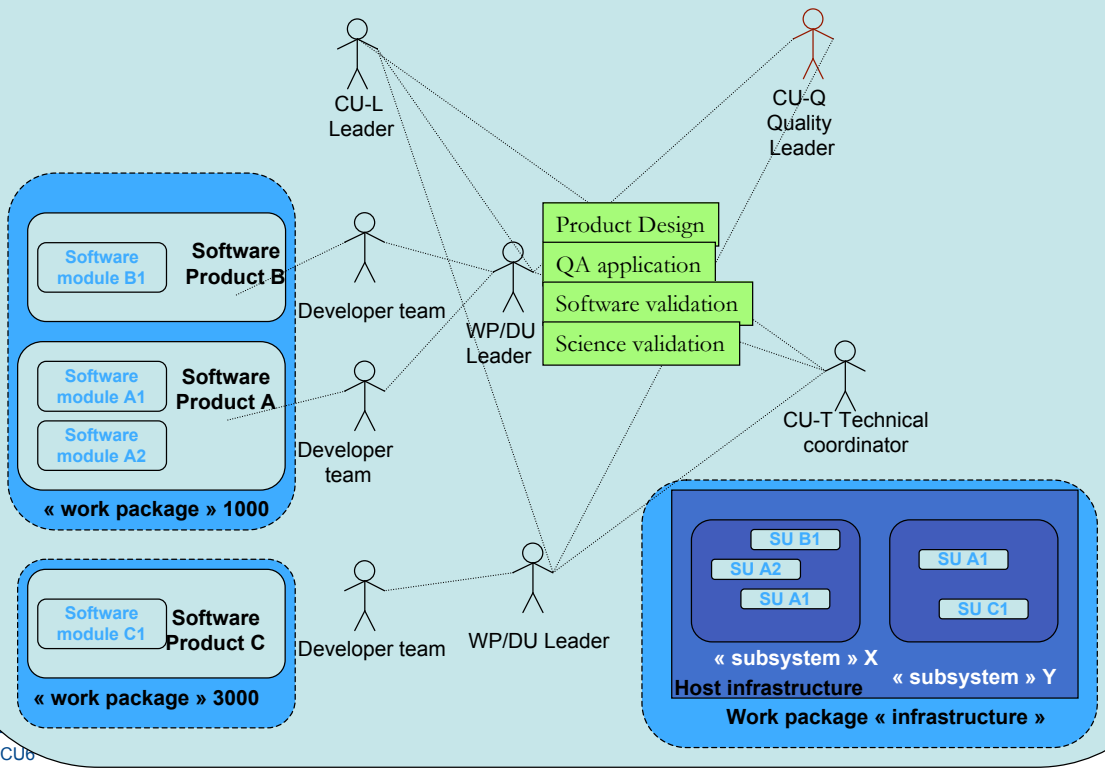

WP decision level

# CU - Actors and roles: CU-L

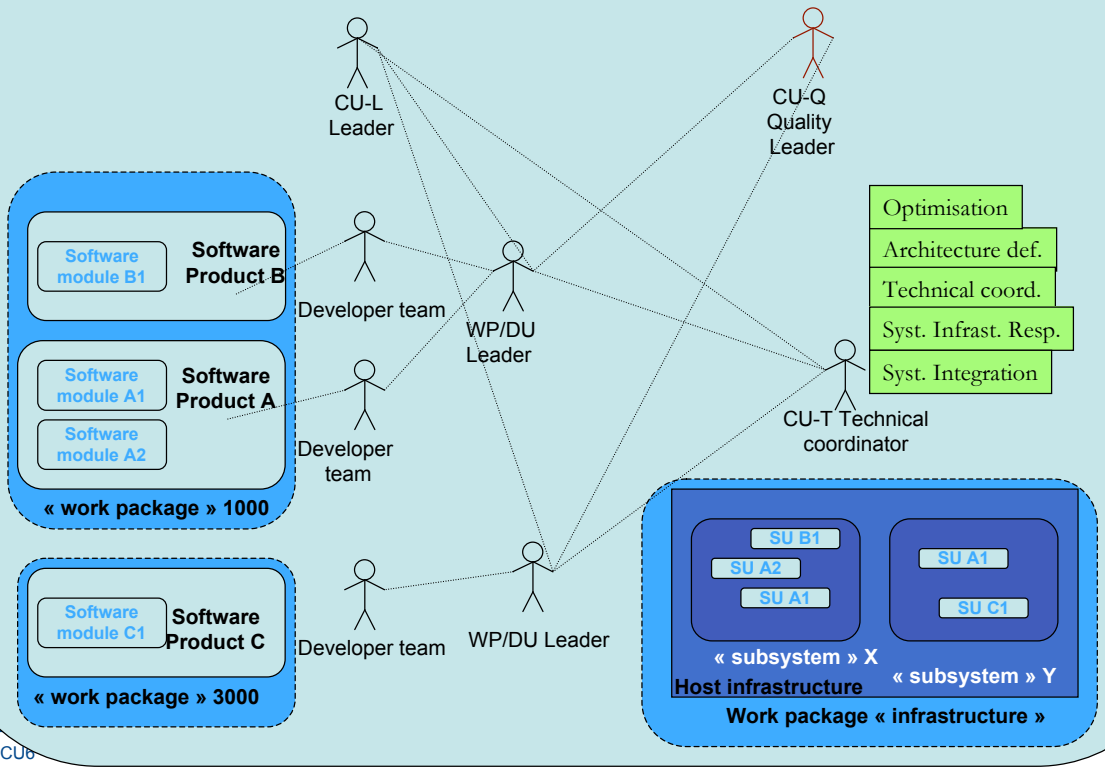





# CU - Actors and roles : DU/WP-L

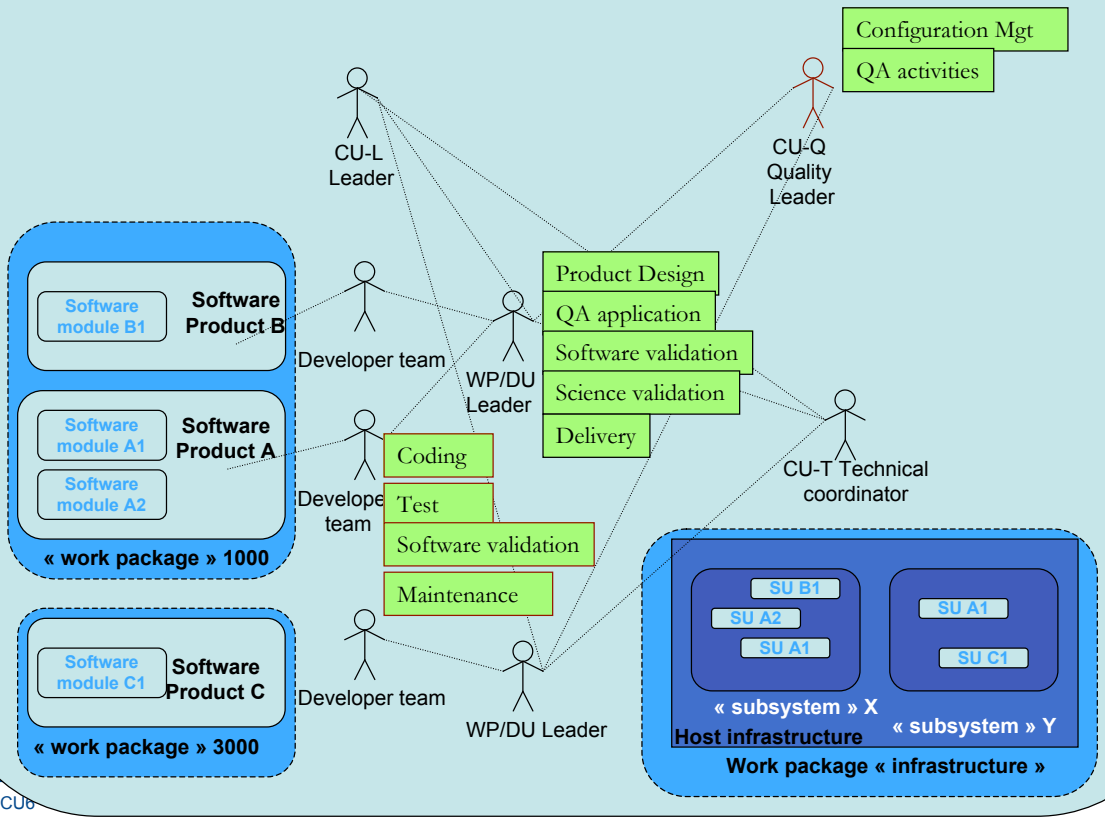


# CU - Actors and roles: CU-T






# CU : Actors and roles : DU+ CU-Q



cnes

CENTRE NATIONAL D'ÉTUDES SPATIALES

# SOFTWARE PRODUCT DEVELOPMENT

cnes

## Development life cycle

✓ The software product development cycle is iterative (1 cycle = 6 months).

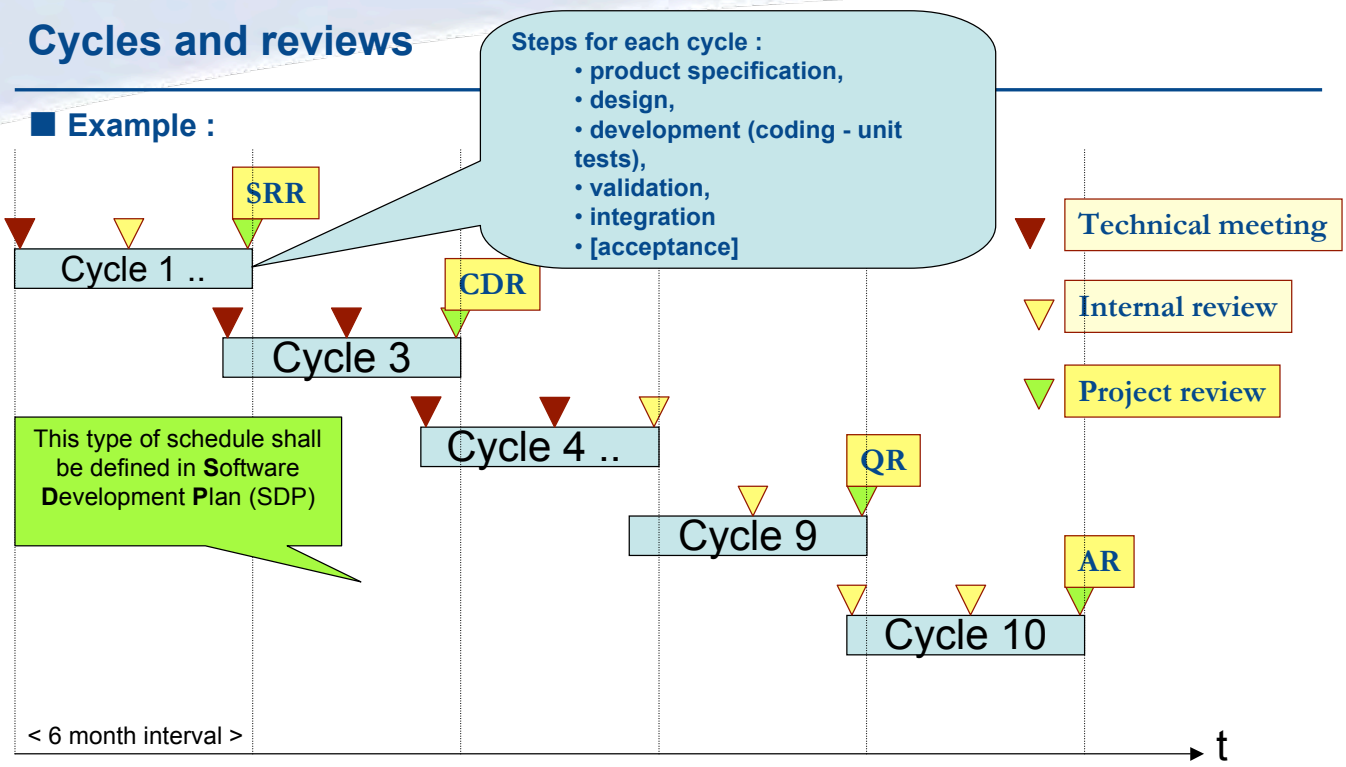
✓ All CUs are synchronised.

✓ Each cycle includes the following activities:

- product specification
- design
- development (coding - unit tests)
- validation
- integration (when expected)
- acceptance (last cycle only)

## Cycles and reviews

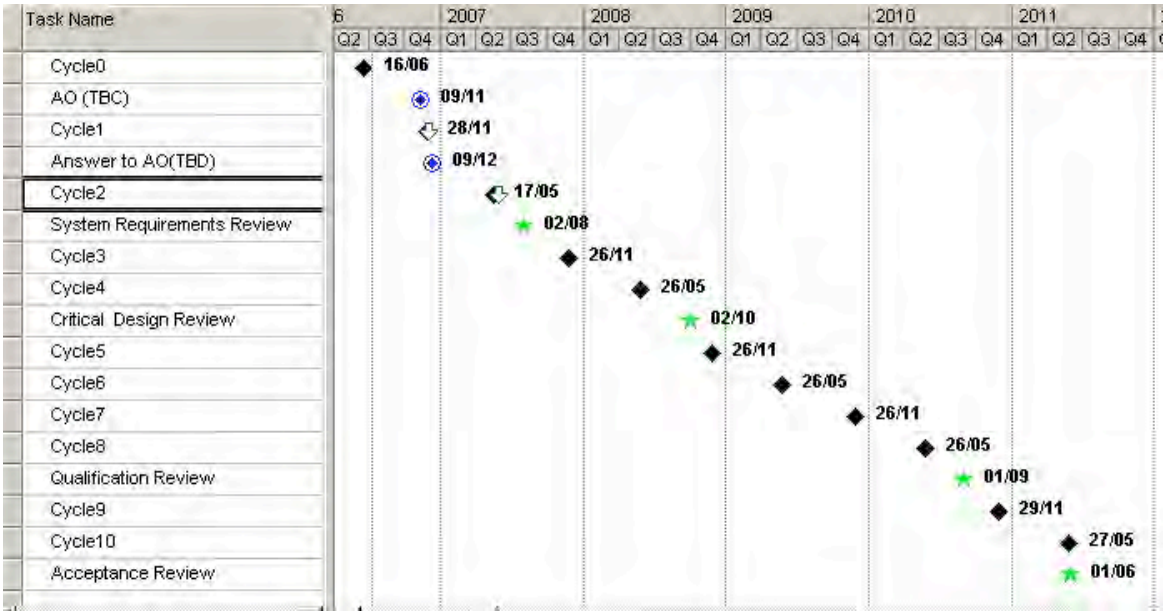
■ **Example :**



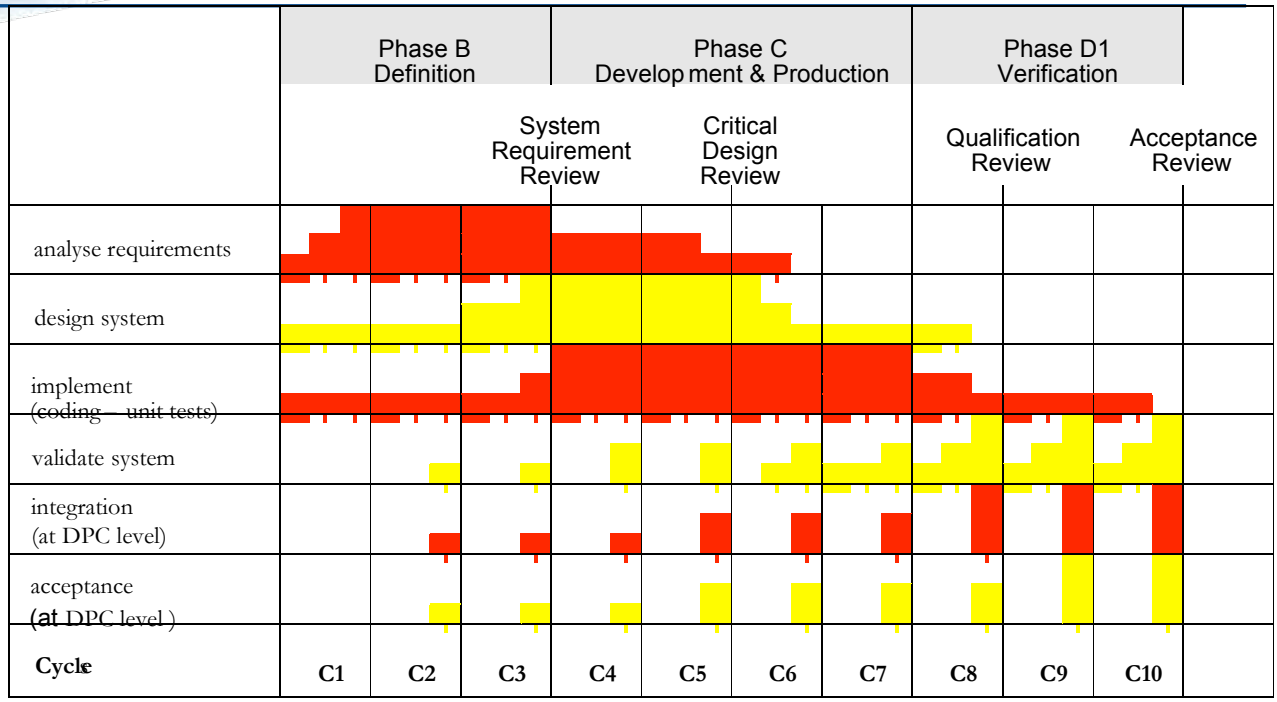



## Project reviews

■ **Proposed schedule of reviews:**



## Main activities during cycles

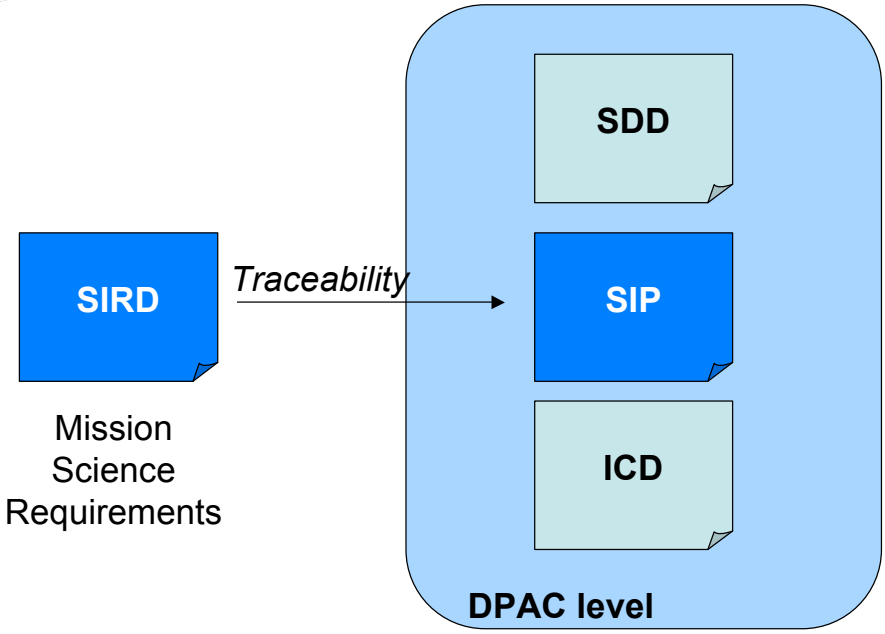





## PAED : Documentation at DPAC level

| Title | Editor(s) / Level | Description / Content |
|---|---|---|
| Science Implementation Plan (SIP) | All CU-L / DPAC | How DPAC will fulfil Science Requirements |
| Interfaces Control Document (ICD) | CU1 / DPAC | Description of interface requirements at Gaia reduction system level |
| Software Design Document (SDD) | CU1 / DPAC | Overall architecture at Gaia reduction system level |

## PAED : Main technical documentation at DPAC level






## PAED : Documentation at DPC level

| Title | Editor / Level | Description / Content |
|---|---|---|
| Software User Manual (SUM) | CU-T / DPC | User manual at host software framework level. |
| System Test Plan (STP) | CU-T / DPC | Description of all the tests for the DPC, including integration test of science products software. |
| System Test Verification Report (STVR) | CU-T / DPC | Verification of all software within the CU or DPC.<br>Results of tests execution (defined in STP), Matrices of requirements verification result (points to test references or manual analysis or …. for each requirement). |
| Configuration Item List (CIL) | CU-Q / DPC | Definition of which Hardware and Software items are to be configuration controlled (Configuration Item) |
| Configuration Baseline (CB) | CU-T / DPC | For Configuration Item: the versions of the technical documents defining the software system (SRS, SDD). For Hardware CI this is the technical specification and configuration of the Hardware and any supporting software system (OS, COTS) |

## PAED : Documentation at CU level

| Title | Editor / Leve(s)l | Description / Content |
|---|---|---|
| Software Requirements Specification (SRS) | CU-T,L / CU | Functional and non functional software system requirements at CU level |
| Interfaces Control Document (ICD) | CU-T,L / CU | Description of interface requirements at CU level (between software products) |
| Software Design Document (SDD) | CU-T/CU | Architectural and Software Design at CU level |



## PAED : Documentation at CU level

| Title | Editor(s) / Level | Description / Content |
|---|---|---|
| Software Development Plan (SDP) | CU-L,T,Q / CU | Description of planning, cycles, roadmap of releases, internal milestones, deliverables for each activity, specific software product assurance and engineering standards and techniques not covered by the PAED, etc. |
| Performance Report Document (PRD) | CU-T/WP-L / CU | Results of estimations and benchmark measures. |
| Software Product Assurance Report (SPAR) | CU-Q/ CU | Results of quality measurement, verification and control |

## PAED : Documentation at DU / WP level

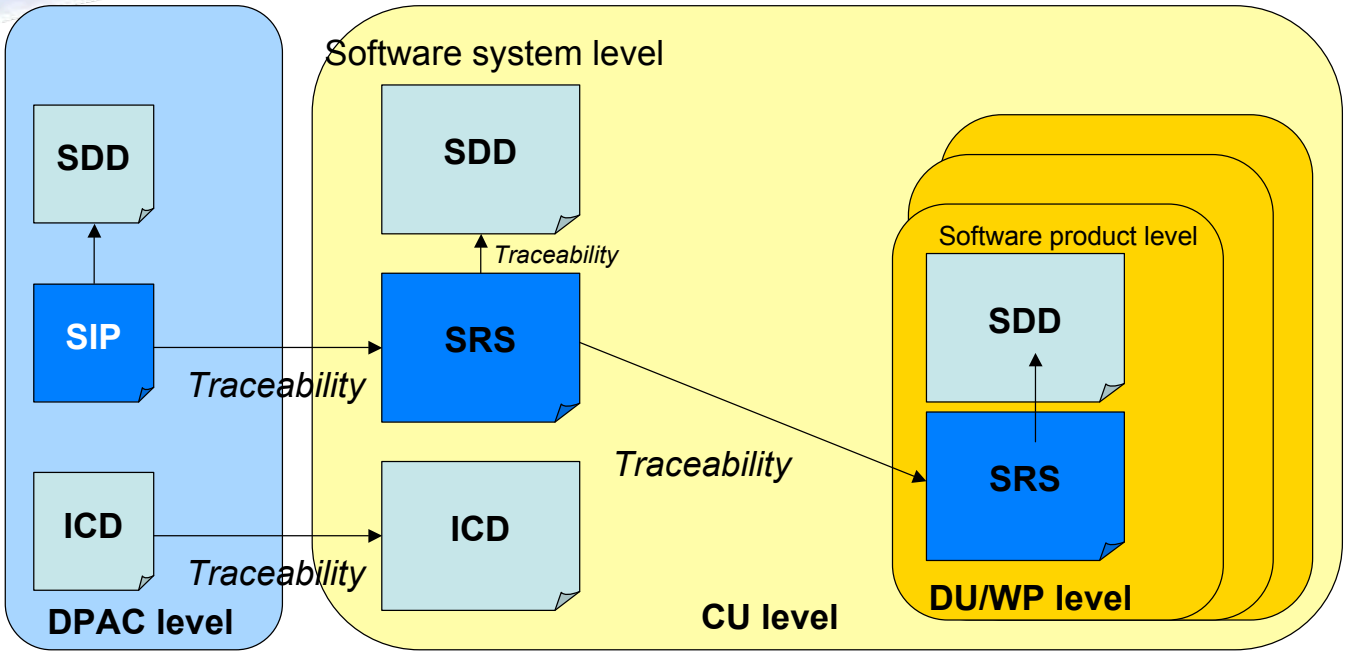
| Title | Editor(s) / Level | Description / Content |
|---|---|---|
| Software Requirements Specification (SRS) | DU-L / DU | Functional and non functional software requirements at DU/WP level |
| Software Design Document (SDD) | DU-L / DU | Design of software products and modules at DU/WP level |



## PAED : Documentation at DU / WP level

| Title | Editor(s) / Level | Description / Content |
|---|---|---|
| Software Test Plan (STP) | DU-L / DU | Test (unit, integration and validation) coverage goals, strategy, etc. end test procedures |
| Software Test & Verification Report (STVR) | DU-L / DU | Test (unit, integration and validation) results<br>Verify all requirements have been fulfilled (for each requirement, demonstrate that there is a test reference or an analysis reference, or etc.) |
| Software User Manual (SUM) | DU-L / DU | Software user manual at software product level |
| Software Release Note (SRN) | DU-L / DU | Description of a software release (identification, modifications, dependences, instructions, etc.) |

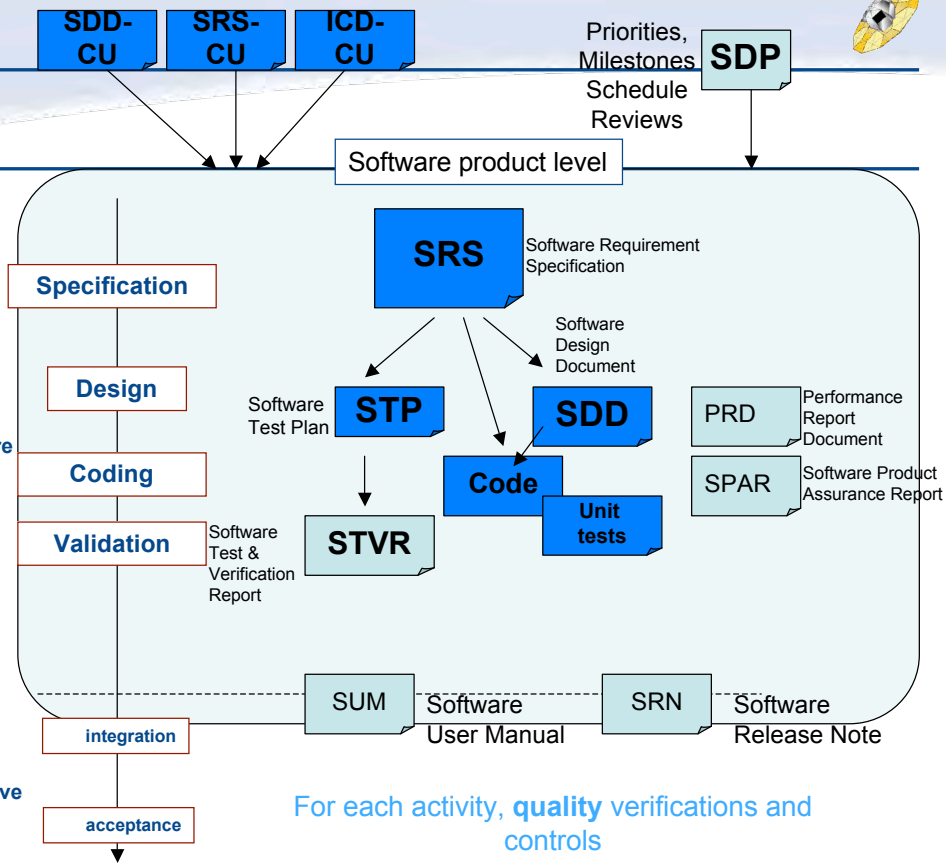
## PAED : Main technical documentation at CU and DU / WP level






## Summary…

- A specification is a set of functional or not functional requirements
- A design is a response to the specification:
  - responses to requirements (defined in SRS + ICD)
  - a definition of how the software is built
- Coding / unit tests
  - Create/Update Code
  - Unit and software integration testing
  - Define how to check that each requirements is verified (Software Test Plan, analysis, etc. )
- Validation
  - Verify that all requirements have been fulfilled (SVM + STR + PRD)




For each activity, **quality** verifications and controls

| Activity Document | Kick-off meeting | Analyse Requirements | Design system Implement | Validate | Integration Acceptance |
|---|---|---|---|---|---|
| SDP | U | U<br>Or F if last cycle | | | |
| SRS | U | U<br>Or F if last cycle | | | |
| ICD | | U | U<br>Or F if last cycle | | |
| SDD | | | U<br>Or F if last cycle | | |
| STP | | U | U | U<br>Or F if last cycle | F |
| STR | | U | U | U | |
| SUM | | | U | U | F |
| SVM | | | | U<br>At last cycle | F |
| PRD | | | U | F<br>last cycle | |
| SPAR | U | U | U | U | U |
| SRN | | | U | U | U<br>Or F if last cycle |



## CU6 : Internal reviews for cycle 2 (Proposal)

- **Kick off meeting : Workshop at Brussels (12-13 Oct. 2006) with all the developers.**
- **Intermediate meeting of the steering committee in mid-cycle to examine progress of :**
  - **SRS documents at DU level (WP from 610 to 650)**
  - **SRS document and ICD at CU level (WP 602)**
  - **SDD and SDP documents for DUs in charge of the development of the first algorithms (Cycle 2 deliverables)**

  **In order to point out the different issues and actions to be done.**
- **Meeting by end of the cycle 2 : Workshop at Toulouse (Date TBD) with all the developers to :**
  - **Examine the work performed during the cycle (based on documentation reviewing, quality checks, …) and expose feedbacks from the Cnes DPC following the delivery of codes and their integration / test on the Cnes infrastructure.**
  - **Prepare the objectives of the next cycle 3.**

## CU6 : Documentation at CU level for cycle 2

- Software Development Plan (v1.x) ;
- Software Requirement Specifications document at CU6 level (v1.0) ;
- Software Design Document at CU6 level (v1.0) ;
- Interface Control Document (v1.0);
- Performance Report (v1.0) ;
- Meeting minutes (workshop, phone conference, …)



## CU6 : Documentation at DU / WP level for cycle 2

- **According to the CU6 SDP (GAIA-C6-SP-OPM-DK-003-1, 18 Sept 2006) :**
  - **Software Requirement Specification**
    - First version (1.0) for the software products identified for the cycle 2 (“7 algorithms” – see table 5) and for the CU6 host software infrastructure.
    - Draft version for the others identified in the SDP (see table 6) and for the CU6 host software infrastructure.
  - **Software Design Document**
    - First version (1.0) for the software products identified for the cycle 2 (“7 algorithms” – see table 5)
    - Draft version for the others identified in the SDP (see table 6) and for the CU6 host software infrastructure.
  - **Software Test Plan document**
    - First version (1.0) for the software products identified for the cycle 2 (“7 algorithms” – see table 5)
    - Draft version for the others identified in the SDP (see table 6) and for the CU6 host software infrastructure.
  - **Codes and STVR (Software Test & Verification Report)**
    - First version (1.0) for the software products identified for the cycle 2 (“7 algorithms” – see table 5).
  - **User manual**
    - First version for the software products identified for the cycle 2 (7 algorithms” – see table 5).

cnes

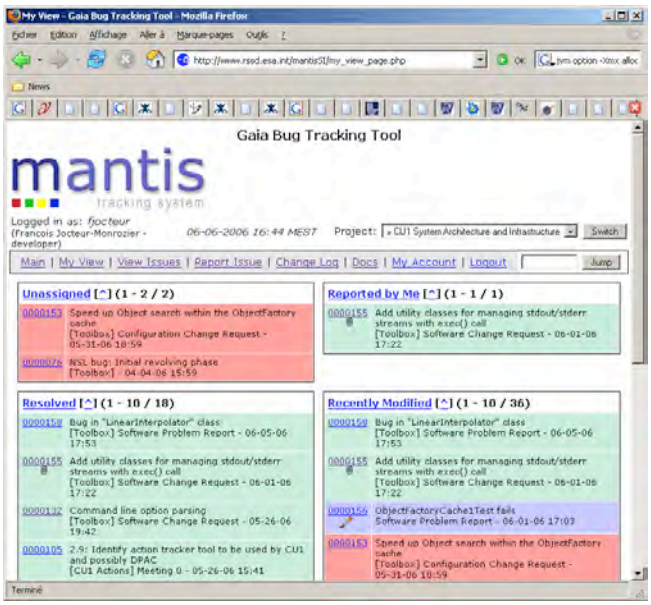
CENTRE NATIONAL D'ÉTUDES SPATIALES

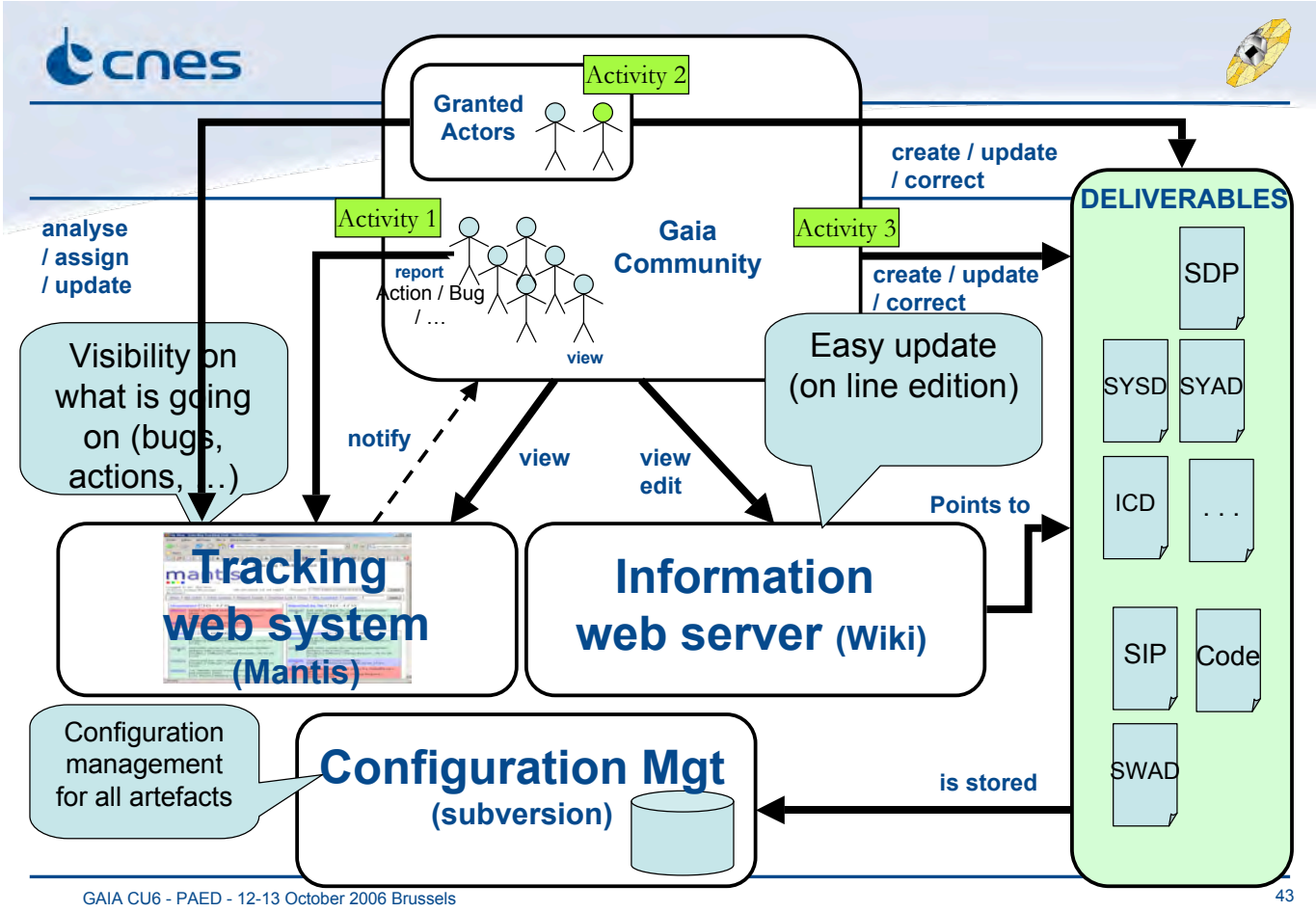
# TOOLS

cnes

## Project and Software Management tools

- MS Project for planning & resources
- Mantis : « Issue » tracker (web based)
  - Action
  - Bug
  - Configuration / Doc changes
  - Software Features
  - Observation report
- Wiki
- Subversion








## « Centralised information » to manage the project

- Everything that concerns several actors should be traced within Mantis.
- For each « issue » use the appropriate category (see Mantis tool) :
  - Action
  - Bug report
  - Configuration change request
  - Document change request
  - Observation report
  - Software problem report
  - Software change report
- Before reporting, check that the issue does not already exist
- Give the associate information to enable an efficient analysis / achievement (Documentation, data, code, …)

## Resources : starting with GAIA DPAC

**GaiaWiki**
www.rssd.esa.int/SA-general/Projects/GAIA/wiki
(must be declared member of a CU by the CU leader to get a login)

**Organisation of DPAC**
**DPACE link**

**General Tools used by DPAC**
**Shared Resources link**

- Documentation
- Parameter data base
- Gaia Glossary
- My Portal
- Email distribution lists
- Synthetic Spectral Database
- Action tracking
- Bug-tracking / Mantis
- Subversion

**software design and technology**
**CU1 link**

- Gaia Toolbox
- System Architecture
- Product Assurance And Engineering Dispositions
- Software Development
- Documentation Standards

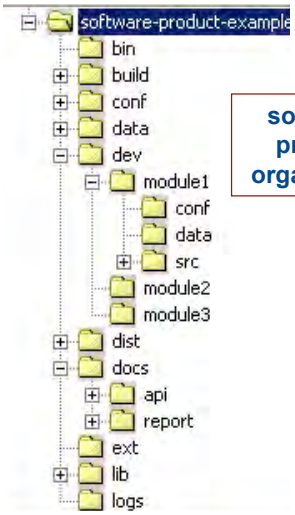
→ **Software Development Toolkit**



## Development environment (see Java workshop)

- Java as programming language
- Subversion (hosted by ESAC)
- [IDE : Eclipse (used for Java Workshop]
- Product tree organisation
- Ant build files with common targets
  - init / check / prepare / compile / [run] / dist / build
  - docapi / test / [metrics] / control / [publish] / [deploy] / clean
- CruiseControl : for automatic control and web publishing
- Additional tools for quality metrics and code checking (PMD, JDEPEND, METRICS, ...)

→ **Software toolkit**




**software product organisation**

## Development environment

- Setting your development environment with the “Software toolkit” :

http://www.rssd.esa.int/SA-general/Projects/GAIA/wiki/index.php?title=CU1:_Software_Development_ToolKit



CENTRE NATIONAL D'ÉTUDES SPATIALES

# PRODUCT QUALITY

## Coding rules objectives

- Java Coding Standard and Guidelines for DPAC, GAIA-C1-SD-ESAC-WOM-005-1, Monday 19th December, 2005

- The compliance with the rules of a coding standard during the phases of development and maintenance of a software allows in particular:
  - To control certain risks of introduction of latent defects,
  - To ensure homogeneity and coherence in the source code.

- These two points make it possible to improve the level of maintainability, reliability of the software.



## Product Metrics measured on the source code

- Method level
  - Method Lines of Code (max 30)
  - Block depth (max 5)
  - Cyclomatic Complexity (max 10)
  - Number of Parameters (max 5)
- Class level
  - Number of Attributes (max 12)
  - Number of methods (max 20)
  - Depth of Inheritance Tree (max 5)
- Software product level
  - Number of Classes
  - Number of Packages

## Product Metrics objectives

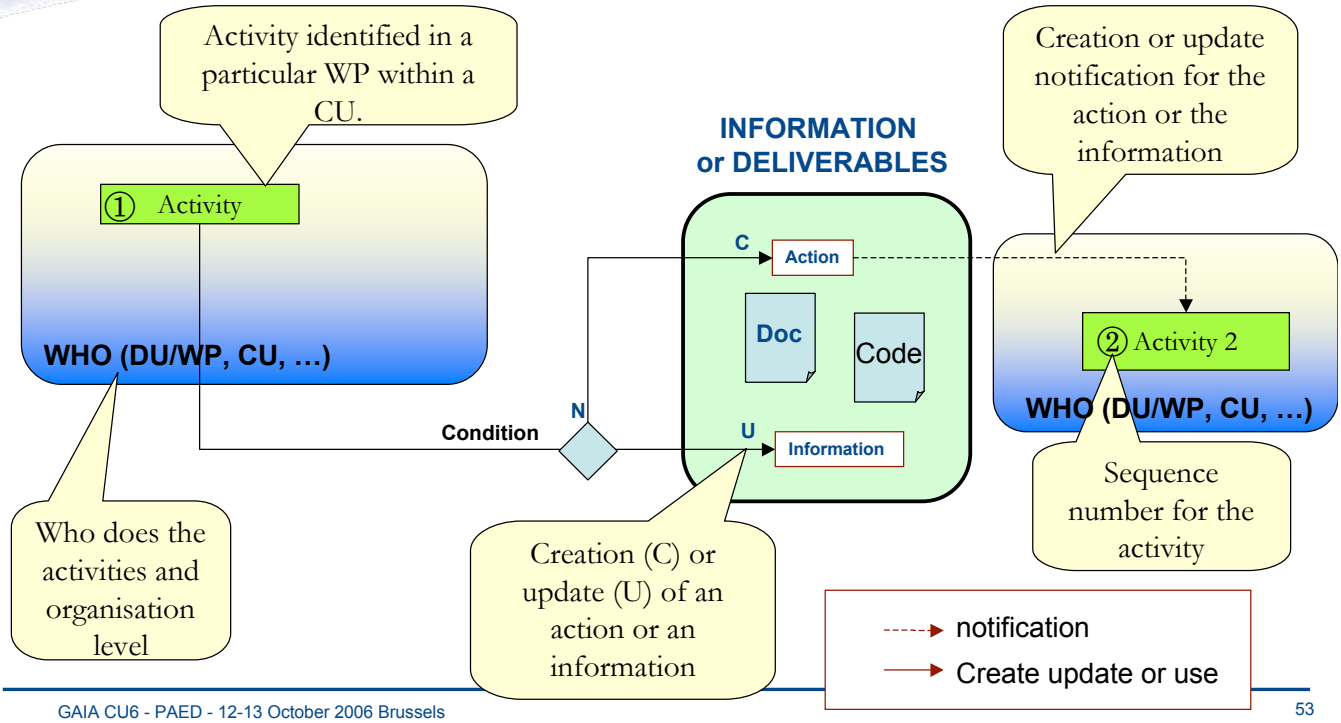
- Product Metrics measured on the source code are complexity metrics.
- The complexity of the source code is used as an indicator of its level of maintainability :
  - a low level of complexity increases the facility of analysis, facility of modification, stability or facility of test of a software component.
- To evaluate the level of maintainability of a software component, measurements of complexity are taken on the source code of this component.
  - Each measurement is then compared with the threshold given in the PAED or in the SDP.
  - The higher the proportion of measurements not respecting the threshold is, the more component maintenability can be considered as difficult.



CENTRE NATIONAL D'ÉTUDES SPATIALES

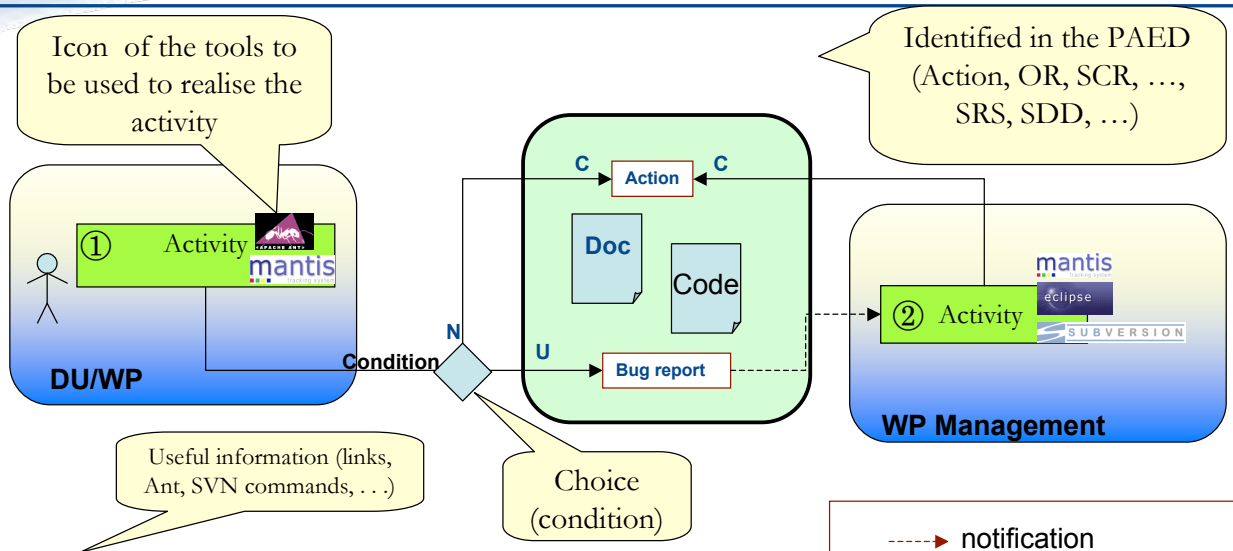
# GUIDE TO APPLY PAED : SOME PROCEDURES / SCENARIOS PROVIDED

## How to read a scenario description (1/3)






## How to read a scenario description (2/3)




**Usefull links & commands:**

**SVN:** http://gaia.esac.esa.int/dpacsvn/DPAC/

**Wiki:** http://www.rssd.esa.int/SA-general/Projects/GAIA/wiki/index.php?title=Data_Processing

**Mantis:** http://www.rssd.esa.int/PHP/mantisGAIA

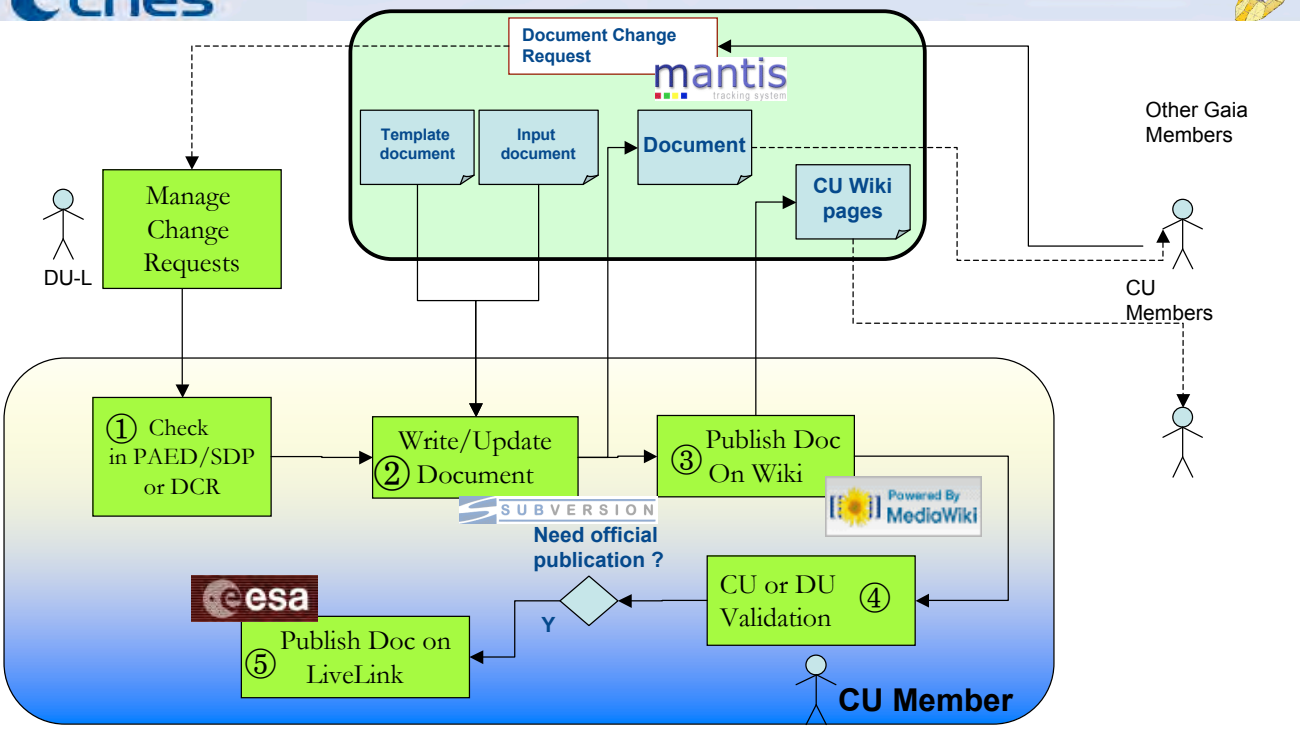
```
<product-directory>%>ant [–f yourbuildfile] build
```

## Scenario list

- **WRITE OR UPDATE A DOCUMENT**
  - ex : Make an ICD
  - ex : Write a SRS at DU Level
  - ex : Write a SDD
- **REPORTING & MANAGING ISSUES**
- **WRITE A NEW SOFTWARE MODULE**
- **REQUEST AN OPTIMISATION**
- **DEVELOP A CLASS, TEST AT UNIT LEVEL**
- **VALIDATION, INTEGRATION AND ACCEPTANCE**
- **PREPARE A PROJECT REVIEW**
- **ORGANIZE AN INTERNAL REVIEW**
- **CHECK YOUR CODE**
- . . .

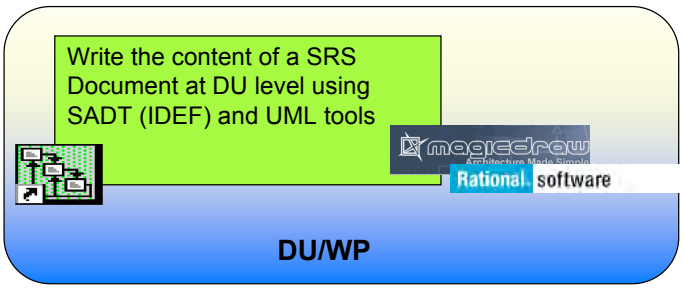


## WRITE OR UPDATE A DOCUMENT



## Write or update a SRS at DU Level

**DEMO**




SADT (french version): http://www-ic2.univ-lemans.fr/~alissali/Enseignement/Polys/GL/node50.html/



CENTRE NATIONAL D'ÉTUDES SPATIALES

# CONCLUSION

## For all CUs, keep in mind

- This document can be improved (Using mantis « Document Change Request »)
- Dispositions are provided to homogenize work practices within the Gaia data processing ground segment project
- The PAED has to be approved by DPACE
- The PAED is adapted for each CU in the SDP



## End …

Thanks for your attention

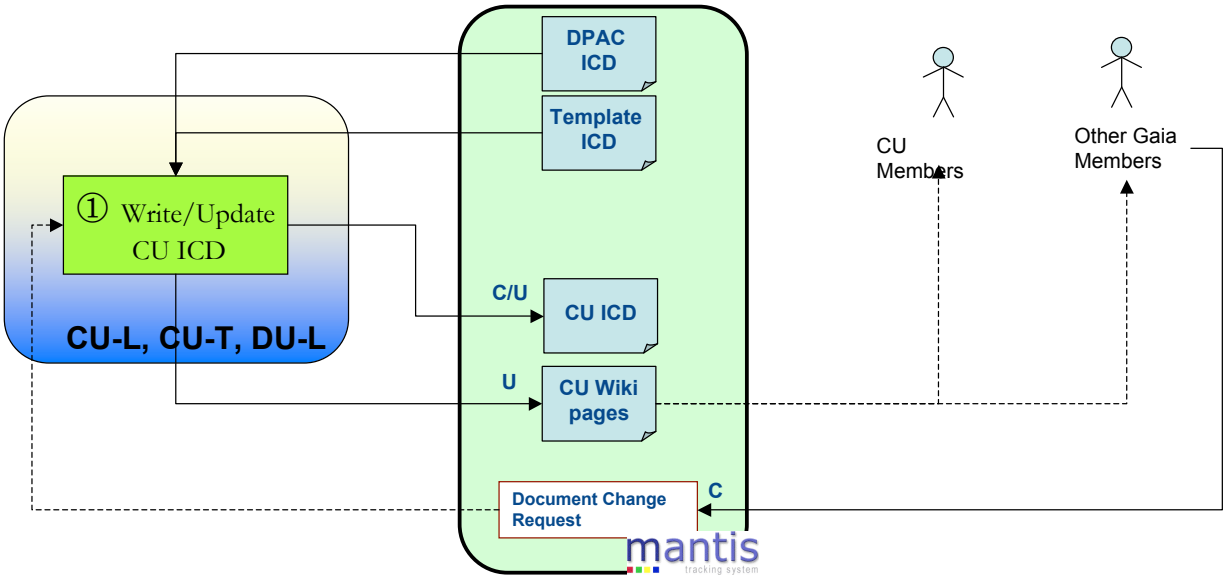
## Scenario list

- WRITE OR UPDATE A DOCUMENT
  - ex : Make an ICD
  - ex : Write a SRS at DU Level
  - ex : Write a SDD
- REPORTING & MANAGING ISSUES
- WRITE A NEW SOFTWARE MODULE
- REQUEST AN OPTIMISATION
- DEVELOP A CLASS, TEST AT UNIT LEVEL
- VALIDATION, INTEGRATION AND ACCEPTANCE
- PREPARE A PROJECT REVIEW
- ORGANIZE AN INTERNAL REVIEW
- CHECK YOUR CODE
- . . .

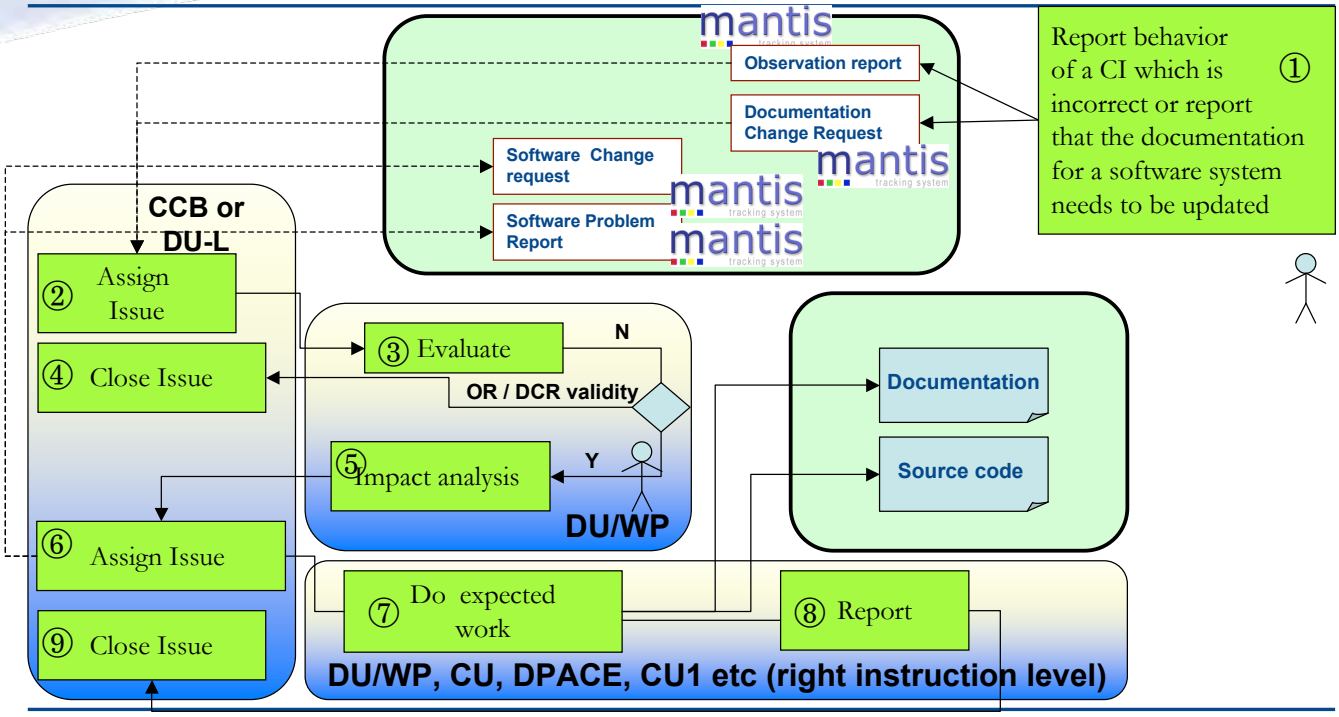


## Make an ICD (CU-L, CU-T, DU-L)

- **Trigger : kick-off cycle 1**

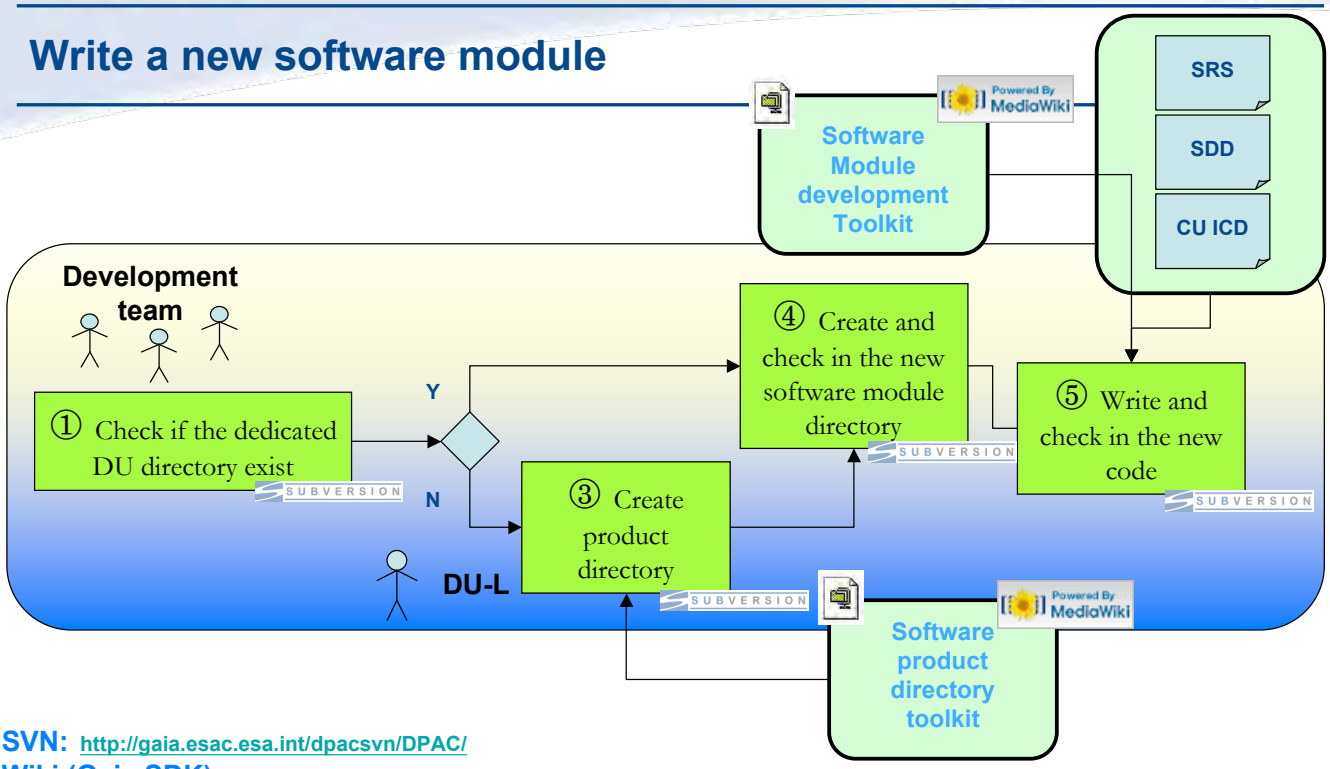


## Reporting and Managing Issues at DU or CU level





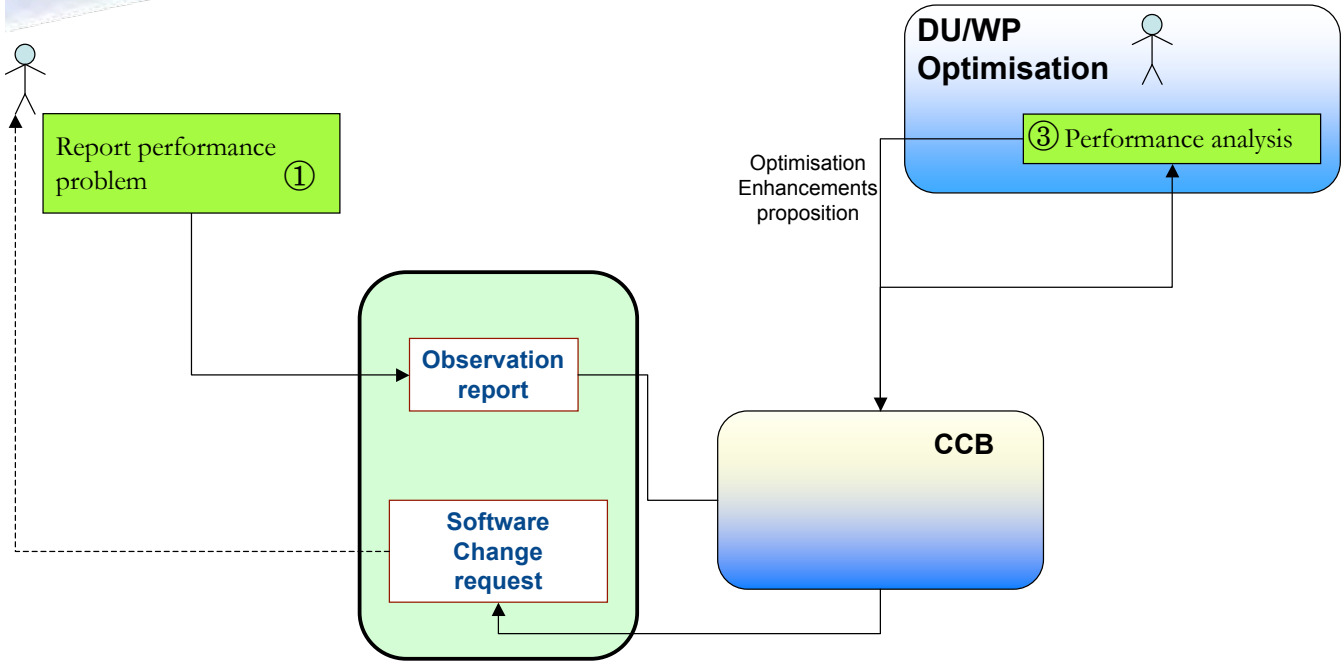

## Write a new software module




**SVN:** http://gaia.esac.esa.int/dpacsvn/DPAC/

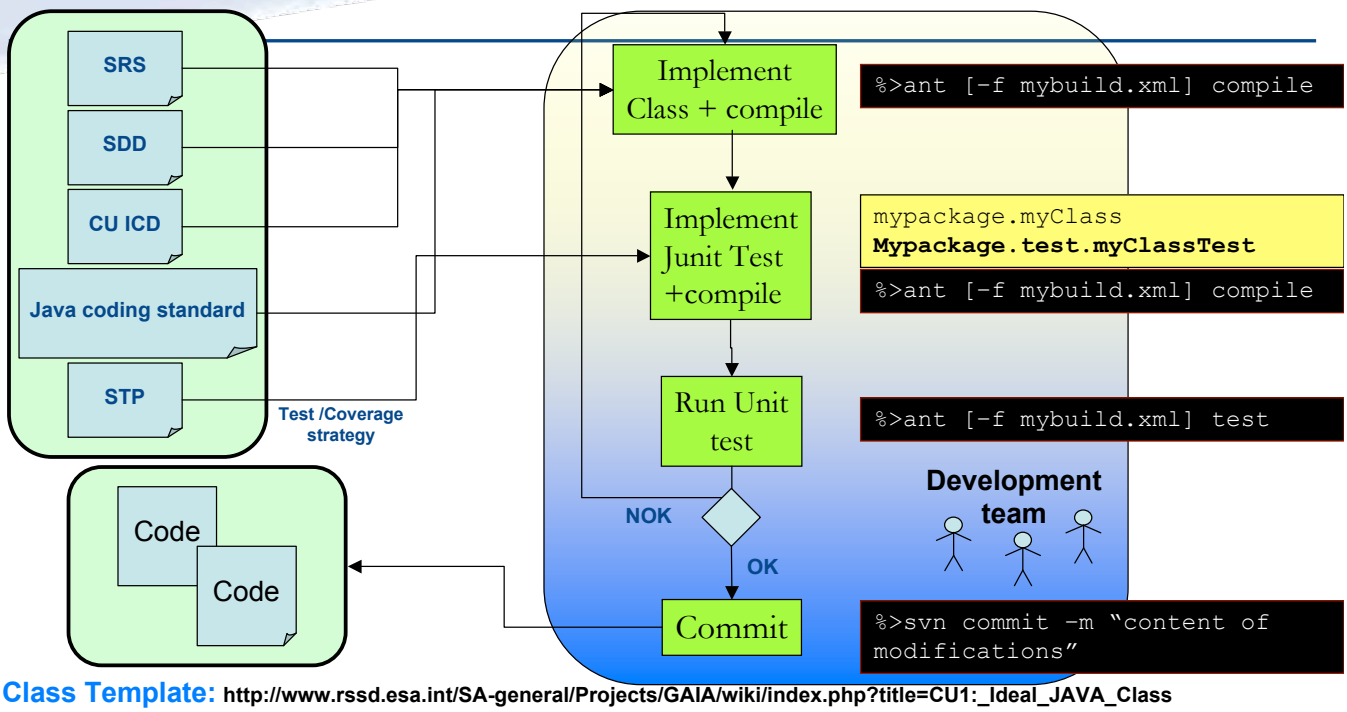
**Wiki (Gaia SDK):** http://www.rssd.esa.int/SA-general/Projects/GAIA/wiki/index.php?title=CU1:_Software_Development_ToolKit

## Request an optimisation





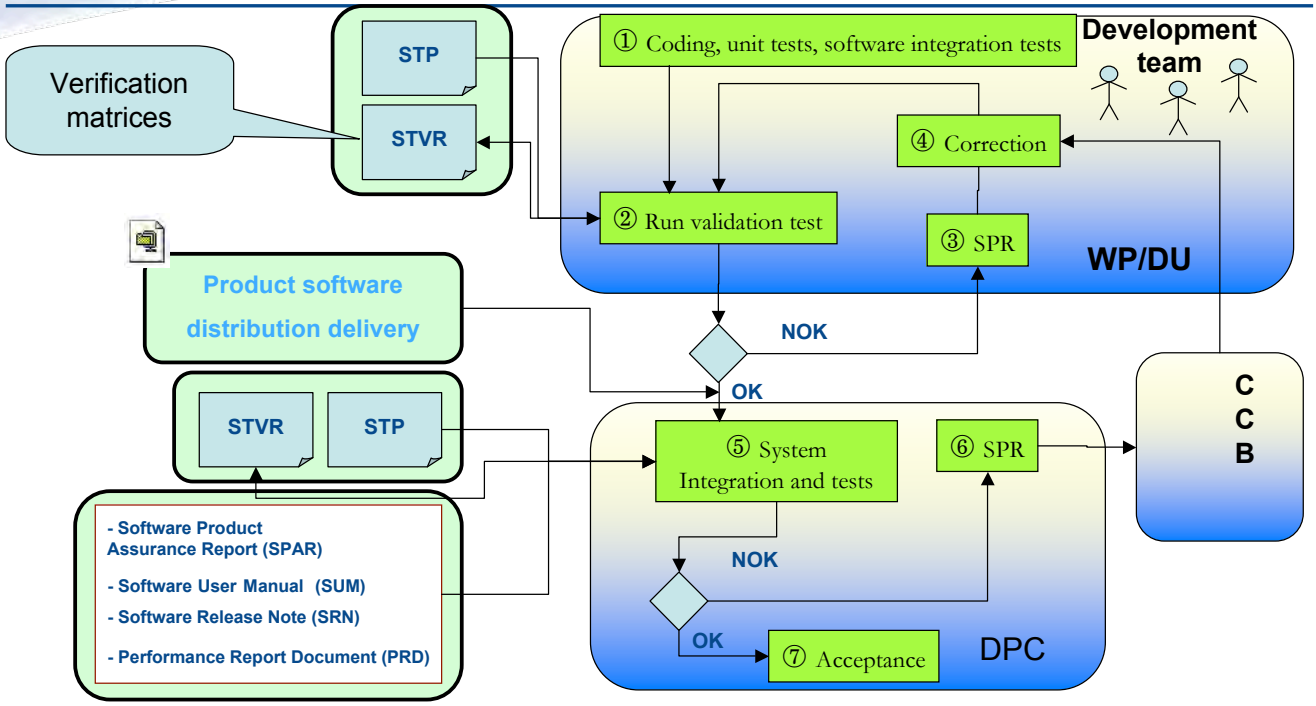

## Develop a class, test at unit level




**Class Template:** http://www.rssd.esa.int/SA-general/Projects/GAIA/wiki/index.php?title=CU1:_Ideal_JAVA_Class

**Junit info:** http://www.rssd.esa.int/SA-general/Projects/GAIA/wiki/index.php?title=Java06:_Testing

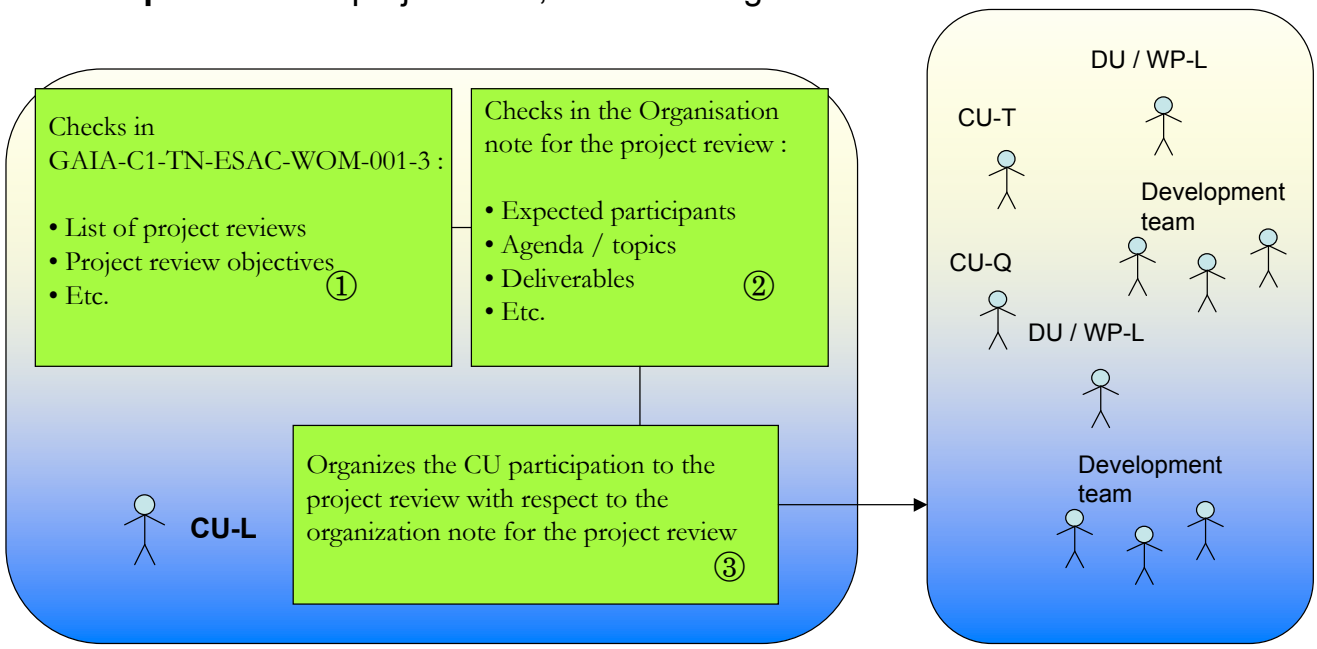
## VALIDATION , INTEGRATION AND ACCEPTANCE





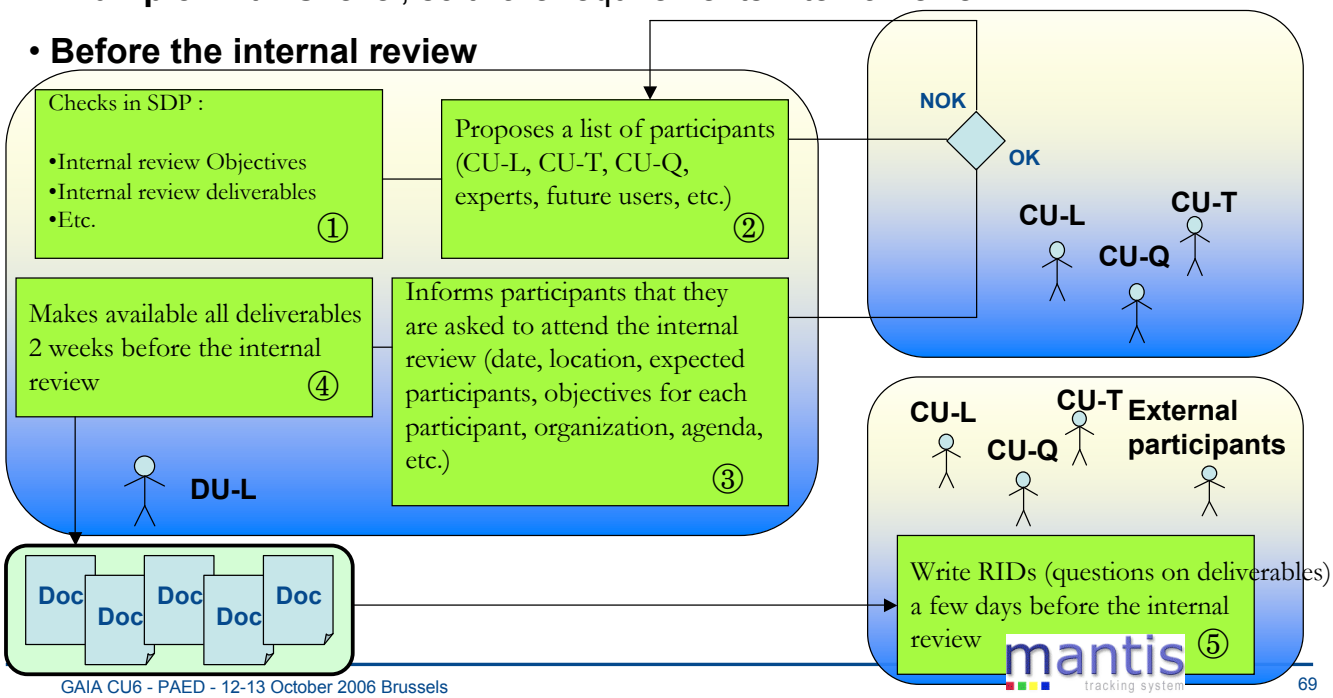

## PREPARE A PROJECT REVIEW

- **Example :** At Gaia project level, Critical Design Review



## ORGANIZE AN INTERNAL REVIEW (1/2)

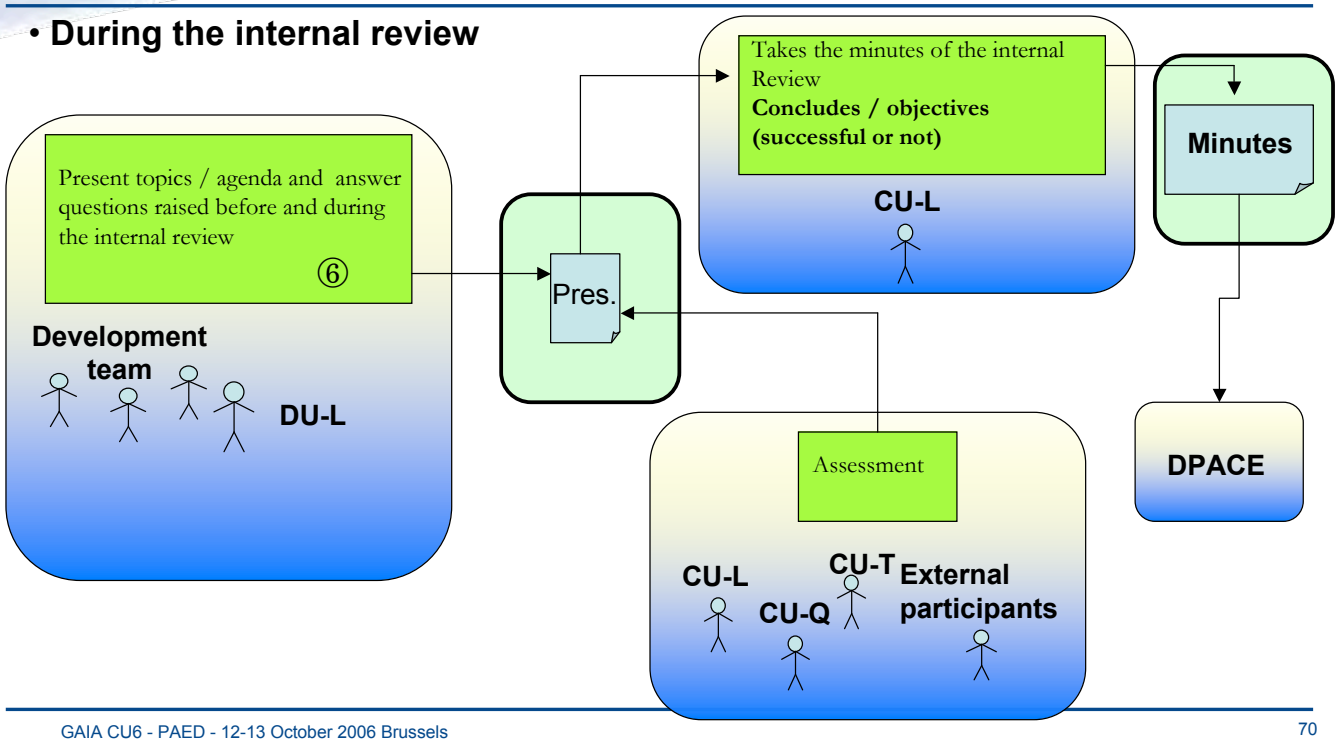
- **Example :** At DU level, software requirements internal review
- **Before the internal review**





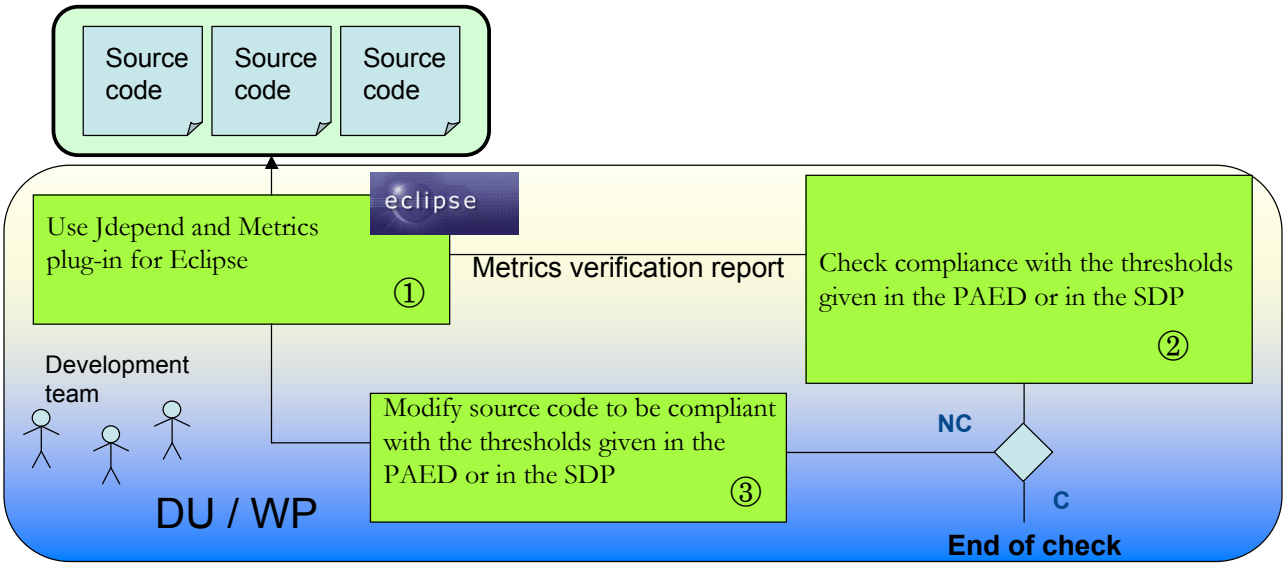

## ORGANIZE AN INTERNAL REVIEW (2/2)

- **During the internal review**



## CHECK YOUR CODE (1/5)

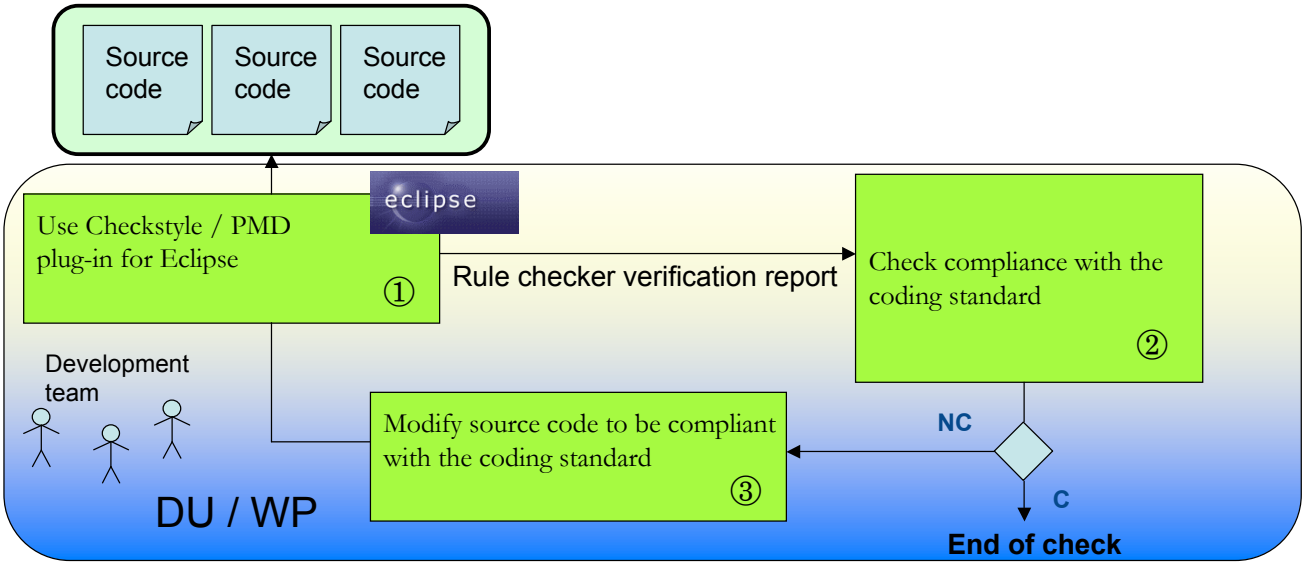
- **Metric measurement**
- **During coding phase, directly by the development team**





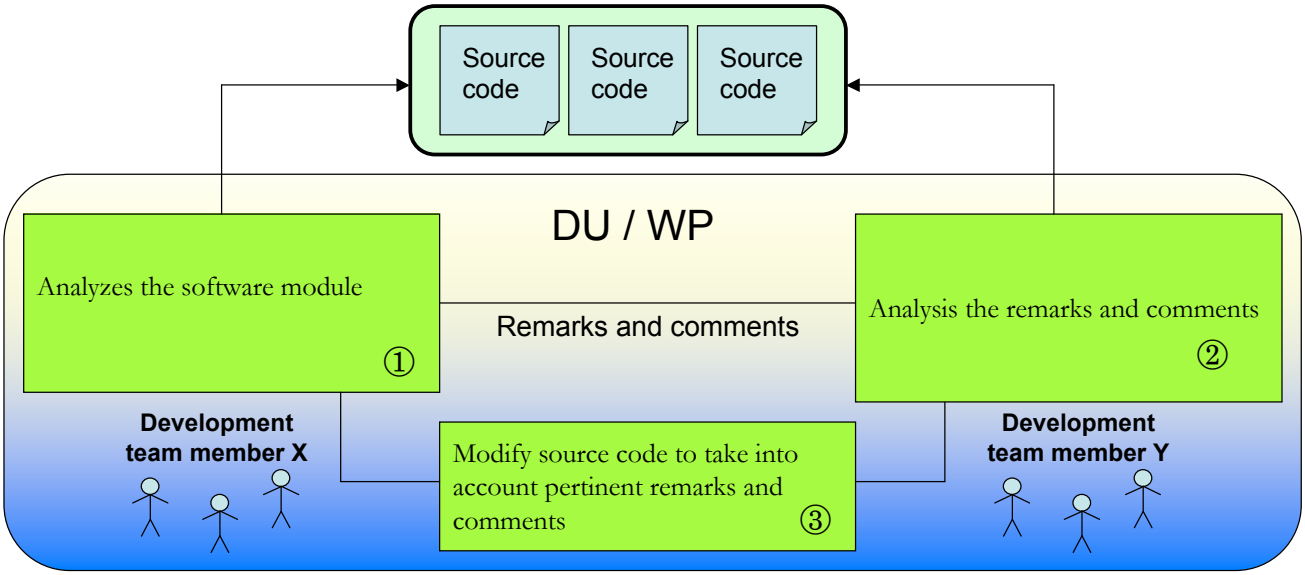

## CHECK YOUR CODE (2/5)

- **Rule checking**
- **During coding phase, directly by the development team**



## CHECK YOUR CODE (3/5)

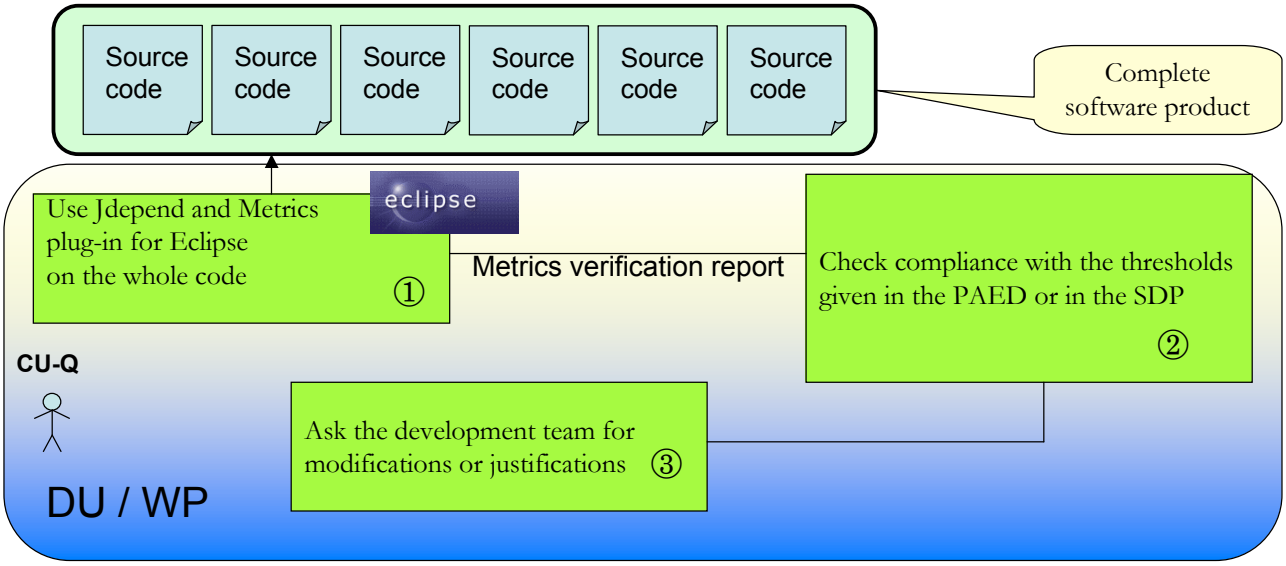
- **Cross check reading**
- **During coding phase, directly by the development team**






## CHECK YOUR CODE (4/5)

- **Metric measurement**
- **Before validation (for each cycle), by the CU-Q**



# CHECK YOUR CODE (5/5)

- **Rule checking**
- **Before validation (for each cycle), by the CU-Q**