FIRST

DPU/ICU On Board Software

Product Assurance Plan

Document Ref.:

Issue: Draft 2

Prepared by: Anna Maria Di Giorgio Date: 08/05/00

Checked by: Riccardo Cerulli-Irelli
Approved by: Renato Orfei

1 INTRODUCTION 6

1.1 PURPOSE OF THE DOCUMENT 6
1.2 ACRONYMS AND ABBREVIATIONS 6
1.2.1 Acronyms 6
1.3 REFERENCES 7
1.3.1 Applicable Documents 7
1.3.2 Reference Documents 8
1.4 OVERVIEW OF THE DOCUMENT 8

2 MANAGEMENT 9

2.1 SOFTWARE LIFE CYCLE 9
2.2 TASKS AND ACTIVITIES 10
2.2.1 User requirements definition phase 10
2.2.2 SR and AD phase 11
2.2.3 Detailed Design and coding phase 11
2.2.4 Software Testing and Delivery phase 11
2.2.5 OBS Maintenance phase 11
2.3 PRODUCT ASSURANCE RELATIONSHIPS 12

3 DOCUMENTATION 12

4 STANDARDS, PRACTICES, CONVENTIONS AND METRICS 13

4.1 DOCUMENTATION STANDARDS 13
4.2 DESIGN AND CODING STANDARDS 13
4.2.1 Change logs and version control 15
4.3 TESTING STANDARDS 15
4.3.1 Unit tests 15
4.3.2 Integration tests 16
4.3.3 Acceptance tests 16

5 REVIEWS AND AUDITS 16

6 TESTS 17

7 PROBLEM REPORTING AND CORRECTIVE ACTIONS 17

8 TOOLS TECHNIQUES AND METHODS 18

9 CONFIGURATION MANAGEMENT 18

9.1 IDENTIFICATION FOR CODE 19

APPENDIX A 20

APPENDIX B 21

APPENDIX C 22

C.1 WORK PACKAGE 1.1 – SPACECRAFT I/F 22
C.2 WORK PACKAGE 1.2 – SUBSYSTEMS I/F 23
C.3 WORK PACKAGE 1.3 - OBS CONTROLLER 24
C.4 WORK PACKAGE 1.4 – DATA PACKETISER 25

C.5 WORK PACKAGE 1.5 – HEALTH AUTONOMY FUNCTIONS....26
C.6 WORK PACKAGE 1.6 – AVM ISSUE....26
C.7 WORK PACKAGE 1.7 – PFM ISSUE....27
C.8 WORK PACKAGE 1.8 – DOCUMENTATION....27
C.9 WORK PACKAGE 1.9.1 – SUPPORT ACTIVITIES: VIRTUOSO OS....28
C.10 WORK PACKAGE 1.9.2 – SUPPORT ACTIVITIES: TEST SOFTWARE....29

Document Status Sheet:

Document Title: DPU/ICU On Board Software PA Plan			
Issue	**Revision**	**Date**	**Reason for Change**
Draft 1		12 Jan 2000	First draft
Draft 2		08 May 2000	Second draft

Document Change Record:

No change record is included for the changes in the draft versions.

Document Title: DPU/ICU On Board Software PA Plan	
Document Reference Number:	
Document Issue/Revision Number: Draft 2	
Section	**Reason For Change**

1 Introduction

This document specifies the product assurance organisation and program plan for the DPU-ICU On Board Software for the three instruments onboard the ESA FIRST/PLANCK mission.
The definition of on board software adopted in this document includes various items related to the generation of an executable program:

- Source code
- Executable code
- Data files
- Documentation

All the tools and the interfaces' simulators developed for the testing activities shall be considered as an integral part of the on board software as well.

1.1 Purpose of the document

This plan intends to define the functions and procedures used to establish the software quality assurance for all the DPU/ICU on board software development life-cycle. It is written to ensure that the software will be developed according to the ESA standards.
As suggested in the Guide to Applying the ESA software Engineering Standards to Small Software Projects (ESA BSSC (96) 2, Issue 1), this document will include the Software Project Management Plan, the Software Configuration Management Plan and the Software Quality Assurance Plan.

1.2 Acronyms and Abbreviations

1.2.1 Acronyms

AD	Architectural Design
ASCII	American Standard Code for Information Interchange
ATP	Acceptance Test Plan
AVM	Avionic Model
CASE	Computer Aided Software Engineering
CDMS	Command and Data Management System
CNR	Consiglio Nazionale delle Ricerche
CPU	Control Processing Unit
DDD	Detailed Design Document
DPU	Digital Processing Unit
EEPROM	Electrically Erasable Programmable Read Only Memory
EGSE	Electronic Ground Support Equipment
ESA	European Space Agency
FIRST	Far InfraRed and Submillimeter Telescope
HIFI	Heterodyne Instrument for FIRST
HK	HouseKeeping
HW	HardWare
IBDR	Instrument Baseline Design Review
ICD	Interface Control Document

ICDR	Instrument Critical Design Review
ICU	Instrument Control Unit
IHDR	Instrument Hardware Design Review
IFSI	Istituto di Fisica dello Spazio Interplanetario
ISVR	Instrument Science Verification Review
NCR	Non Conformance report
OBS	On-Board Software
PACS	Photoconductor Array Camera and Spectrometer
PA	Product Assurance
PDR	Preliminary Design Review
QA	Quality Assurance
RID	Review Item Discrepancy
SCCS	Source Code Control System
SCR	Software Change Request
SPIRE	Spectral and Photometric Imaging REceiver
SPR	Software Problem Report
SPU	Signal Processing Unit
SR	Software Requirement
SSD	Software Specification Document
SVVP	Software Verification and Validation Plan
SW	SoftWare
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
UR	User Requirement
URD	UR Document
WP	Work Package

1.3 References

1.3.1 Applicable Documents

Document Reference	Name	Number/version/date
AD1	PA Requirements for FIRST/Planck Scientific Instruments	PT-RQ-04410 Issue 1 September 1997
AD2	Guide to Applying the ESA Software Engineering Standards to Small Software projects	ESA BSSC (96) 2 Issue 1
AD3	ESA Software Engineering Standards	ESA PSS-05-0 Issue 2
AD4	FIRST/Planck Instrument Interface Document Part A	PT-IID-A-04624 Draft 0 30 Sept 1997
AD4-H	FIRST/Planck Instrument Interface Document Part B Instrument “HIFI”	PT-HIFI-02125. 1 August 1999
AD4-P	FIRST/Planck Instrument Interface Document Part B Instrument “PACS”	PT-PACS-02126. 04 Dec. 1999
AD4-S	FIRST/Planck Instrument Interface Document Part B Instrument “SPIRE”	PT-SPIRE-02124 Issue 0.2 01 Aug. 1999

AD5	Guide to Software Verification and Validation	ESA PSS-05-10 Issue1 February 1994
AD6	HIFI PA Plan	SRON-U/HIFI/PL/1999-008 Issue 2, 18-02-2000
AD7	SPIRE PA Plan	Draft 1, 05-02-1998
AD8	PACS Project PA Plan	Draft, 26-11-1999
AD9	FIRST S/C to Instruments dedicated Transfer Layer Protocol Specification	TBW

1.3.2 Reference Documents

Document Reference	Name	Number/version
RD1	The C (ANSI C) Programming language. (Kernighan, B.W., Ritchie, D.M.)	1989, Prentice Hall PTR

1.4 Overview of the document

TBW

2 Management

The IFSI management structure is shown in the organisation chart given in Figure 2-1: it is the same for the three Consortia. The IFSI local Project manager is responsible for both the DPU/ICU HW and SW development. As far as the OBS is concerned, his tasks are:

- to define the development schedule necessary to meet the Consortia milestones;
- to manage and monitor the activities of the development team;
- to manage technical and schedule risks.

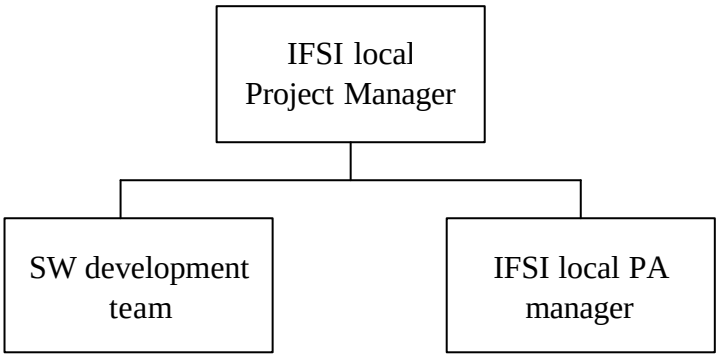


Figure 2-1

The SW development team is responsible for providing all the software products (from AVM release to Flight model release) and documentation. The same team will carry out all the IFSI internal testing activities (see sections 4.3) and will assist all the instrument integration activities involving software use and testing.
The PA manager is responsible for the implementation of the PA activities related to the project. He is located at IFSI and reports directly to the FIRST Instruments PA Project managers.
The PA manager has in charge the overall configuration management of the DPU/ICU OBS ensuring that the software configuration status list is prepared and maintained.

2.1 Software life cycle

The project life cycle represents all the activities related to the design, development, operations and maintenance of the software. It is therefore structured as a model which organises the project activities into phases and defines what activity occurs in which phase.
Since the products of a software development project will be delivered in a timely manner, it is possible to provide a planning of the SW development phases with an indication, for each of them, of all the activities to be performed and all the items to be delivered.
According to ESA Standards for Small Software Projects (ESA BSSC (96) 2, Issue 1), the life cycle of the DPU/ICU OBS development is assumed to be divided into the following phases (see Figure 2-2):

- user requirements definition
- software requirements definition and architectural design

- software detailed design and coding
- software testing and delivery

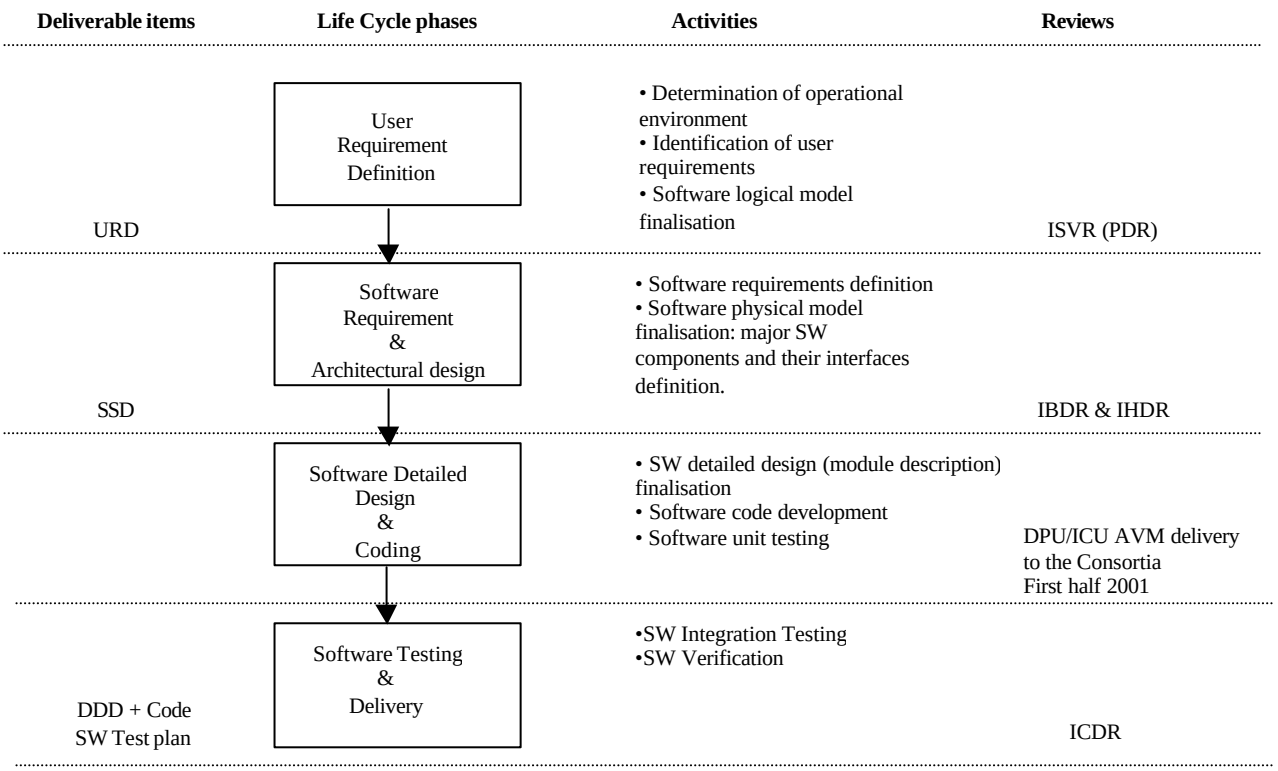


Figure 2-2 Life cycle of the DPU/ICU OBS development

2.2 Tasks and activities.

The present PA Plan is referred to all DPU/ICU OBS development activities described in the development plan reported in Appendix C. In the following the activities belonging to each development phase are briefly described.

2.2.1 User requirements definition phase

The user requirements shall be defined in close co-operation with the three Instruments' teams and the URD document shall be written. An implementation-independent model of what is needed by the user (SW logical model) shall be provided.
The PA activities in this phase will be to ensure the final integrity and completeness of the URD and to check its compliance with the applicable documents and with the methods and standards defined in the applicable documents.

2.2.2 SR and AD phase

The SR shall be defined and the SW Architectural Design shall be finalised. The SW Specification Document will be provided (including the SW architecture and describing the method adopted for the design, see AD2) and the traceability of the AD components versus the SR will be checked. The traceability of the SR versus the UR will be checked as well: for each item of the SW requirements list, the reference UR will be indicated.
The PA activities in this phase will be to ensure the final integrity and completeness of the SSD, to check its compliance with the applicable documents and to ensure the full mapping of the requirements on the design components.

2.2.3 Detailed Design and coding phase

The SW code shall be developed according to the standard described in section 4.2. All the unit tests shall be carried out, to check the single module interfaces. The DDD document will be provided containing mainly a commented copy of the source code.
The PA activities in this phase will be to perform all the necessary software configuration management activities: all deliverable code will be identified in a configuration item list. No code inspection activities will be performed.

2.2.4 Software Testing and Delivery phase

All the OBS integration tests will be performed according to IFSI internal verification plans. The System testing activities will be carried out according to the verification plans, see section 4.3.3 .
When all the IFSI internal testing phases are passed, the OBS will be delivered to the Consortia for the acceptance testing phase. At this stage the software will be ready to be programmed into PROMs and installed on the final target system (i.e. EEPROMs).
The PA activities in this phase will be to perform all the necessary software configuration management activities (create and maintain the history logs for software changes, check the completeness of changes, ensure the traceability of all the modifications) and to participate in all the test phases (creating and maintaining the logs for the test reports, the SPRs and the NCRs).

2.2.5 OBS Maintenance phase

Two main deliveries are foreseen during the lifetime of the project: the AVM delivery and the PFM delivery. The bulk of the DPU/ICU OBS will be delivered at the AVM review. The PFM package will be mainly an upgrade of the AVM package, with the modifications of the code requested by changes in the OBS specifications and with new modules included only if the corresponding specification were not available at the AVM delivery.
Therefore, it can be said that the OBS maintenance starts after the AVM delivery.
The maintenance activities between the AVM review and the PFM delivery will be performed by IFSI using the IFSI development environment to provide the new images of the upgraded code. No packetisation of the new images will be performed: it is expected that this will be one of the services provided by the SCOS 2000 package.
IFSI will not develop any OBS Maintenance dedicated facility.

All the configuration management/document upgrade/testing activities on the new code will be done in this phase by IFSI. This approach is considered as applicable to all the Subsystem tests phases. After the PFM delivery, IFSI assumes that the OBS Maintenance will be an ICC task, ensuring all the necessary support to the ICCs for the development of an OBS Maintenance Facility (the writing of the DPU/ICU On Board Software User Manual is part of this support). This software package will have the following main functionalities:

- Possibility to modify the code of all the instruments on-board softwares;
- Possibility to modify all the tables on-board;
- Possibility to generate binary images from the new software (No packetisation of the executables is foreseen);
- To allow comparison of the on-board code with the already stored memory images;
- To provide a graphical user interface in order to ease all the previous functions execution.

2.3 Product assurance relationships

Given the organisation reported in Figure 2-1, in case of disagreement between the PA manager and the SW development team, the problem will be reported to the Project manager for a decision.

No IFSI internal quality inspection audits are foreseen. External audits can be arranged, see Section 5.

3 Documentation

The following documents for the DPU/ICU OBS will de delivered:

- DPU/ICU OBS User requirement Document: common to the three consortia, with a section for each instrument dedicated to the non common user requirements;
- DPU/ICU OBS Software Specification Document: one per each Instrument;
- DPU/ICU OBS Detailed Design Document: one per each Instrument;
- DPU/ICU OBS Product Assurance Plan (this document): common to the three consortia
- DPU/ICU OBS Software Verification and Validation Plan: one per each Instrument.

The reference codes of the documents shall be in accordance with the document management indications provided by the three instruments (see section 9).
Non deliverable documents will be written for IFSI internal use, e.g. technical notes. These documents will not be under configuration control.

4 Standards, Practices, conventions and metrics

4.1 Documentation standards

The structure and contents of all documents will be compliant with the indications in AD2.
Each document will contain header pages showing approval and status of the document.
The final format of the documents will be MSWord 97 and Acrobat PDF.
After a preliminary circulation in a draft form, for gathering all the main comments, the documents will be delivered to the consortia according to the three following procedures:

1. SPIRE: the documents will always be delivered to the ESA DMS domain responsible, who will put them in the DMS.

2. PACS: the documents will be put in the PACS internal web dedicated page.

3. HIFI: the document will be put in the HIFI internal web dedicated page.

4.2 Design and coding standards

Design standards: The software design will be performed according to the standards written in AD2. The Structured Design method will be adopted, with some special extensions for the real time applications architectural designing. The adopted method and the formalism will be described in the SSD.

Coding standards: The software will be written in ANSI C (see RD1), avoiding to use non-standard features. If any non standard features are needed for technical reasons this will be documented. In the final documentation ALL use of non-ANSI features will be documented.

IFSI internal conventions will be:
- Max number of code lines in one file (module): 1000.
- Max number of code lines in a function: 50.
- For each source file a dedicated header file will be provided including constants and variable declarations, type definitions, function declarations, defines etc., for the usage inside the corresponding module. A separate include file containing all the prototypes of the functions that will be needed publicly will be provided, as well as an additional global header file with all the definitions and declarations that will be made accessible from all the modules.
- Each module shall start with a header, showing the title, a brief description, creation date and last update, listing and describing the functions in the module. In the following, two examples of headers are reported below, for a module and a function respectively:

1. Example of module header:

```
/*****************************************************************************
* BEGIN FILE
* File name:
* Version.Revision:
*
* Purpose:
*
* Public Functions:
*
* Private Functions:
*
* Description:
*
* Creation date & author: dd-mm-yyyy              XXX
* Version, Update date & author:    nn.mm         dd-mm-yyyy          XXX
* END FILE
*****************************************************************************/
```

2. Example of function header

```
/*****************************************************************************
* BEGIN FUNCTION
* Function name:
*
*  Purpose:
*
* Description:
*
* Syntax:
*
* Input:
*
* Output:
*
* Return:
*
* Global Variables:
* Files Accessed:
*
* Version, Creation date & author:   nn.mm        dd-mm-yyyy          XXX
*
* Version, Update date & author:     nn.mm        dd-mm-yyyy          XXX
* Update changes:
*
* END FUNCTION
*****************************************************************************/
```

- All private definitions and declarations shall be put at the beginning of the module/function.
- OBS variables naming convention: TBD
- When possible the memory for data structures will be allocated dynamically.
- Every system call will be checked for an error return and the system error text will be included in the error message log. The overall error handling strategy will be described in the SSD (AD section).

4.2.1 Change logs and version control

A SW system undergoes many changes in its lifetime. Revision control can be viewed as *change management*: it allows the developers a degree of control over what changes are made to the system, and allows them to find information about previous changes to the system. In other words, it avoids users making changes to collide with each other.
All the DPU/ICU OBS software deliverables will be under revision control (there will be an internal IFSI revision history, used by the development team in order to have a control on the s/w changes, and an external revision history, keeping the record of the different versions of the delivered packages) as well as all the provided documentation, after the first Issue.

The version control tool that will be used by IFSI is the **Cyclic Software CVS (Concurrent Version System)** freeware. The tool works by keeping a single copy of all the source code (and of all the other deliverables) and putting it into a database of the change history of the developed system: CVS maintains a history of all changes made to each directory tree it manages. Using this history, it will be possible to recreate past states of the tree, knowing when, why and by whom a given change was made.
CVS can be set up as a client/server system: the version history is stored on a single central server and the client machines will have a copy of all the files that the developers are working on.
Finally, the selected tool supports parallel development, allowing more than one developer to work on the same sources at the same time and managing the concurrent changes.
A detailed report on the history of the project can be obtained easily and included in the package documentation provided with each delivery of the code.

4.3 Testing standards

In the following, the definitions in AD2 and AD5 are adopted for the different testing phases.

4.3.1 Unit tests

A ‘unit’ of software is here intended to be a module/file: the inputs to unit testing are the successfully compiled modules from the coding process. These are assembled during unit testing to make the largest units, i.e. the components of architectural design. Tests will be performed to check both the internal logic of the software and the overall software functionality.
IFSI will not provide any unit test documentation (test plan, test input and output data, test reporting forms).

4.3.2 Integration tests

With Integration testing it is here intended the process of testing the developed software against the architectural design: the already unit tested components are integrated to make the system. The tests will be designed to verify the data and control flows across the interfaces between the AD components. Performance tests will be included as well.
Integration tests can be performed on an emulation system.
The integration tests plan will be provided as a draft before the IFSI on site integration. The first issue of the document will be essentially composed of a complete set of the tests reports and will be delivered with the software.

4.3.3 Acceptance tests

The acceptance testing activities are referred to the process of testing the system against the software (and user) requirements. The input to these tests is the successfully integrated SW.
The tests will be designed to verify each functional requirement and each performance requirement to the system. A set of tests will be dedicated to the system interface testing, using the ICD as the baseline of the acceptance. For these purposes, simulators of the interfaces of the DPU/ICU OBS with the spacecraft and the other instruments' subsystems will be developed and used during instrument integration. These simulators will be part of the DPU/ICU OBS maintenance facility, as well as all the analysing tools developed by IFSI and necessary to put in a readable form the data flows to be controlled and to examine the function and performance of the OBS.
The acceptance tests will be performed at the three consortia PIs premises, on the real systems (avionic models/simulators). It is assumed that the EGSEs will be available for the testing activities.
The SVVP will be provided as a separate section of this document, to be completed at the end of the SR & AD phases. It will be part of a new issue of the PA Plan to be delivered before the CDR.

5 Reviews and audits

No internal reviews and audits for pure quality assurance purposes will be performed. During the overall software design and development life cycle, the IFSI SW development team will have periodic internal technical reviews for the purposes of reporting the work package status/problems to both the PM and the PA manager.
Since the code inspection activity is considered as essential by the Consortia, some dedicated "external" audits will be organised with members of the Consortia, in order to review the code before the official delivery. These reviews will be arranged either at IFSI premises or via internet exchange of the relevant documentation. The quality activity will be performed by the Consortia.
The proposed reviews are (see Appendix C for the dates reported in the development plan):
1. OBS Intermediate Review;
2. OBS 2nd Review.

For what concerns the documentation, both the PM and the PA manager will review all the documents before the official delivery (even in the draft form). The documentation review cycle is described in section 7. The Software Specification inspection activities (Software

Requirements review, Architectural Design review) are considered as part of the Documentation life cycle.

In addition to the dedicated reviews, the DPU/ICU OBS software status will be part of the DPU/ICU hardware reviews foreseen in the three hardware Development Plans.

6 Tests

See section 4.3.

7 Problem reporting and corrective actions

- **Documents.**
  All deliverable documents have to run through a review cycle, which is shown in Figure 7-1. The documents are first delivered from IFSI to the Consortia, and then are included into a delivery pack to ESA, in correspondence of the main reviews. Therefore, in principle, there should be two different reviews: one with the Consortia and the second with ESA. The first review will be included into the periodic FIRST OBS meetings organised between IFSI and the Three Consortia. The OBS draft documents and the software design technical notes will be circulated at least one month before the meetings. The comments from the Consortia will be gathered at least one week before the meetings. The comments and the consequent modifications will be discussed at the meetings. This iterative approach will lead to the first issue of the various documents, approved by the Consortia, to be submitted to ESA.
  The formal review cycle of the submitted documentation through RIDs will be applied starting from the first submission of the document to ESA.
  The considered formal ESA reviews are:

Review	OBS documents to be delivered
ISVR	DPU/ICU OBS URD, PA plan
IBDR,IHDR	DPU/ICU OBS SSD
ICDR	DPU/ICU OBS DDD,SVVP

- **Code.**
  Applicable only to the Software Acceptance Review: A Test Review Board shall be constituted, including a representative of the SW development team (and the PM and the PA manager). All problems detected during tests shall be reported to the Review Board, using the SW problem reporting form (SPR). The form will contain a field to indicate the criticality level of the related problem: major problems will be marked as critical. All the helpful material for the problem investigation will be attached to the form (e.g. intermediate steps results, test actions log files etc). The critical SPR's will be reported in the Monthly Progress Report of the DPU/ICU OBS PM to the Instruments PMs.

  The PA manager will provide a history log for all SPR's.

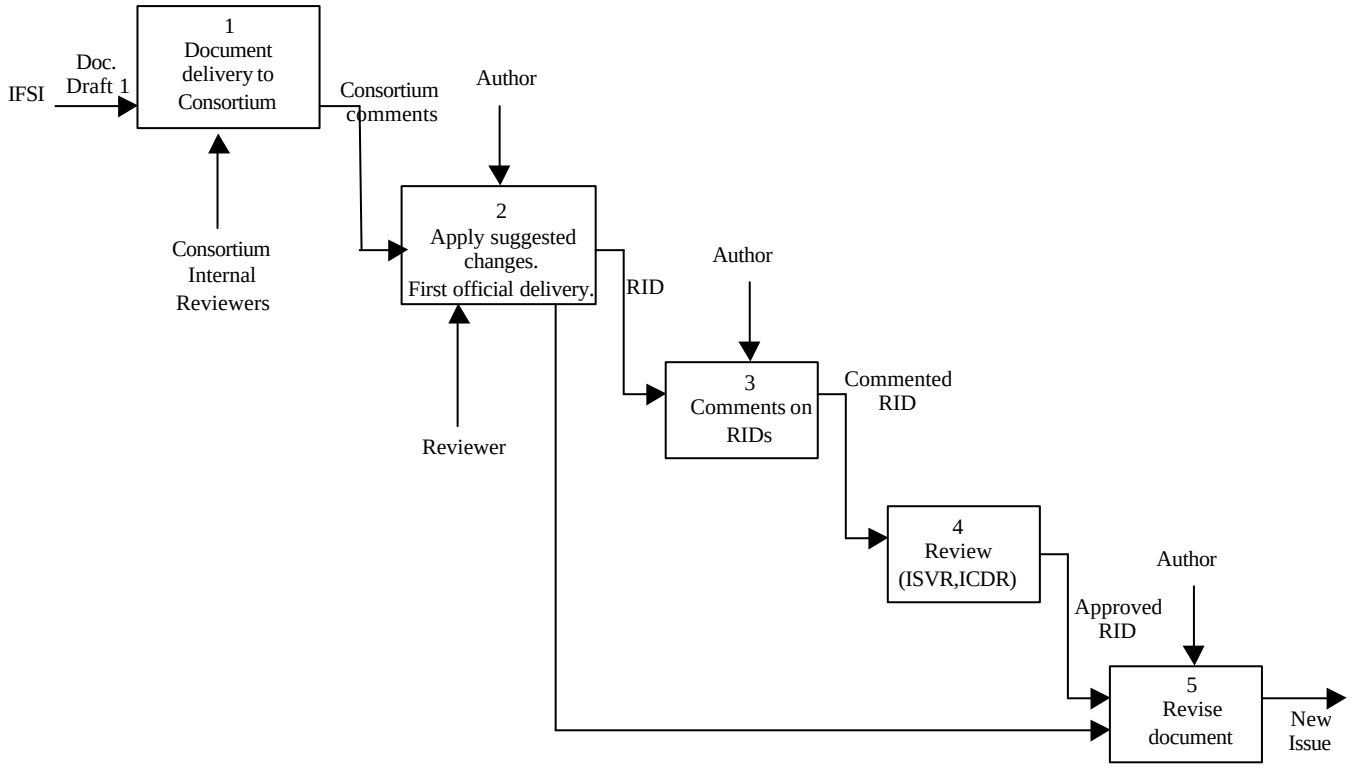


Figure 7-1 Review cycle for documents

8 Tools techniques and methods

A CASE tool will be used for the design phases: a top-down approach will be adopted, both in the definition of the logical model (independent on the HW architecture and based only on the user requirements listed in the URD) and in the architectural design.
The **AxiomSys** CASE tool, based on the Yourdon DeMarco structured analysis method has been selected for these purposes. Given its real time extensions, this tool will allow the production of state transition diagrams and timing diagrams. The references for the tool are:
AxiomSys System Analysis C.A.S.E. tool
Structured Technology group, Inc. (http://www.stgcase.com)

The tool will be used under a Windows NT environment, in a client server configuration.

9 Configuration Management

The responsible for the configuration management is the PA manager (see section 2).

For referencing the documents, even in the case of common documents, IFSI will adopt different documentation identification rules for the three consortia, following the indications contained in:

1. SPIRE: SPIRE Document Management Plan, SPIRE-RAL-DOC-000032, Issue 1.0, Feb 99
2. HIFI: HIFI Document Breakdown (SRON-U/HIFI/PL/1999-007) TBC
3. PACS: PACS Documentation and Identification Procedure, PACS-ME-GP-001, Issue 1, 22-03-2000

9.1 Identification for code

The following file extensions will be used:

Extension	File type
EXE	executable code, linked code
OBJ	Relocatable object codes
C	C source code
H	Include files
HEX	EEPROM/PROM code

The headers of modules and functions have been already described in section 4.2.
In both cases the Version.Revision number of the code shall be present in the header.
The configuration control will be performed according to what written in section 4.2.1.

The tools and the interfaces' simulators developed for the testing activities will not be put under configuration control.

Appendix A

Sample sheet for SPR

TBW

Appendix B

Sample sheet for RID

TBW

Appendix C

Software development schedule

In the following table the schedule of all the activities foreseen for the software development is reported. The start/end dates are taken from the DPU development plan. A copy of the Gantt Chart provided for each consortium is reported in figure C.1.

WP	Start date PACS	End date PACS	Start date HIFI	End date HIFI	Start date SPIRE	End date SPIRE
1 - OBS	01-09-1999	01-10-2003	01-09-1999	01-12-2003	01-09-1999	15/04/2004
1.1 - Spacecraft I/F	01-01-2000	30-03-2001	01-01-2000	30-03-2001	01-01-2000	30-03-2001
1.2 - Subsystem I/F	01-01-2000	30-03-2001	01-01-2000	30-03-2001	01-01-2000	30-03-2001
1.3 - OBS Controller	01-01-2000	30-03-2001	01-01-2000	30-03-2001	01-01-2000	30-03-2001
1.4 - Data Packetiser	01-01-2000	30-03-2001	01-01-2000	30-03-2001	01-01-2000	30-03-2001
1.5 - Health autonomy modules	19-12-2000	07-05-2001	19-12-2000	07-05-2001	19-12-2000	07-05-2001
1.6 - AVM issue	01-05-2001	30-09-2001	01-05-2001	30-11-2001	01-05-2001	20-03-2002
1.7 - PFM issue	21-03-2002	01-10-2003	21-03-2002	01-12-2003	21-03-2002	16-04-2004
1.8 - Documentation	01-12-1999	29-10-2001	01-12-1999	29-10-2001	01-12-1999	29-10-2001
1.9 - Support Activities	01-09-1999	08-01-2002	01-09-1999	08-01-2002	01-09-1999	08-01-2002
1.9.1 - Virtuoso OS	01-09-1999	31-07-2000	01-09-1999	31-07-2000	01-09-1999	31-07-2000
1.9.2 - Test modules	01-06-2000	08-01-2002	01-06-2000	08-01-2002	01-06-2000	08-01-2002

There is one main WP (OBS), shortly described in the following. All the other work packages shall be considered as a breakdown of all the activities related to it. They are described in detail in the following sections.

Title: On Board Software (OBS)

Objectives:
To provide the computer board the functionality to manage the instrument.

WP description:
Development of the DPU/ICU OBS.

- Design activities: user requirements definition, logical model design, software requirements definition, architectural design. (WP 1.3, 1.8)
- Coding activities: detailed design and code development. (WP 1.3, 1.5, 1.8)
- Testing activities: unit tests, integration tests, acceptance tests. CDMS interface simulator development. Subsystems' interfaces simulators development. SVVP definition. Test reports generation. (WP 1.1, 1.2, 1.4, 1.6, 1.7, 1.8, 1.9)

C.1 Work Package 1.1 – Spacecraft I/F

Objective: to develop the software task dedicated to handle the interface between the DPU/ICU and the CDMS. This task will run in parallel to the other OBS tasks: the necessity of a dedicated task with these purposes is due to the fact that the CDMS interface will be compliant with the MIL-STD-1553B standard in which there is only one Bus Controller (the CDMS) and all the activities (in terms of telecommands reception/telemetry sending) are commanded/synchronised by it. The task will implement both the low level electrical interface protocol defined by the MIL-STD-1553B standard and the FIRST dedicated Transfer Layer Protocol (AD9) for managing the communications with the spacecraft.

WP description: This work package will be composed by the following activities:

- task design: logical decomposition
- task design: architectural design
- task design: detailed design-pseudocode
- code development: source C code
- software testing: unit tests
- software testing: integration tests

WP inputs:

- OBS URD (draft, 01/01/2000)
- Spacecraft to instrument ICDs. (01/01/2000)
- IID A (01/01/2000)
- IID B (01/01/2000)
- AD6 (01/06/2000)
- Development platform with: Axiomsys CASE tool (01/01/2000)
  ADSP compatible development board
  Virtuoso Operating system
  C language development environment
  MIL-STD-1553B compatible board
  Configuration management tool
- Simulators of the other OBS components for the task interfaces testing (output of WP 1.4) (when necessary, starting from 01/08/2000)

WP outputs

- Logical model: the sections relevant to this task (11-08-2000)
- Software specifications: the requirements relevant to this task (11-08-2000)
- Architectural design: the sections relevant to this task (11-08-2000)
- Source code (02-04-2001)
- Executable files (either the single executable or the complete image of the OBS, depending on the operating system implementation) (AVM, PFM Issues)

C.2 Work Package 1.2 – Subsystems I/F

Objective: to develop all the software modules dedicated to handle the interfaces between the DPU/ICU and the instruments' subsystems. Even if this software will not be a separate task, the development of the functions dedicated to the subsystem interface handling needs to be included into a separate work package, essentially because of the additional instrument-specific software specifications they need, with respect to the overall requirements. Moreover the optimisation of

the interface reading/writing performances will need dedicated subsystem simulators and test procedures.

WP description: This work package will be composed by the following activities:

- task design: logical decomposition
- task design: architectural design
- task design: detailed design-pseudocode
- code development: source C code
- software testing: unit tests
- software testing: integration tests

WP inputs:
- DPU/ICU to subsystems ICDs.(?)
- OBS URD (draft, 01/01/2000)
- Operating modes/observing procedures detailed descriptions
- Development platform with: Axiomsys CASE tool (01/01/2000)
  ADSP compatible development board
  Virtuoso Operating system
  C language development environment
  Configuration management tool
- Subsystem I/F simulators HW (?)
- Simulators of the other OBS components for the functions interfaces testing (output of WP 1.4) (when necessary, starting from 01/06/2000)

WP outputs
- Logical model: the sections relevant to this task (11-08-2000)
- Software specifications: the requirements relevant to this task (11-08-2000)
- Architectural design: the sections relevant to this task (11-08-2000)
- Source code (02-04-2001)
- Executable files (either the single executable or the complete image of the OBS, depending on the operating system implementation) (AVM, PFM Issues)

C.3 Work Package 1.3 - OBS Controller

Objective: to develop the software task necessary to handle all the commands execution, that is all the activities necessary to the implementation of the instruments operating modes. This task is the core of the DPU/ICU OBS: it will be able to interpret the telecommands and to translate them into single instructions for the instrument. It will be interfaced with the Spacecraft Interface task and the Data Handling task via the standard interprocess communication protocols supported by the Virtuoso operating system: this will ease the task interface testing during the system integration activities.

WP description: This work package will be composed by the following activities:

- task design: logical decomposition
- task design: architectural design
- task design: detailed design-pseudocode
- code development: source C code
- software testing: unit tests

- software testing: integration tests

WP inputs:
- OBS URD (draft 01/01/2000)
- Ground to spacecraft ICD. (01/01/2000)
- Operating modes/observing procedures detailed descriptions (01/01/2000)
- AD6 (01/08/2000)
- Development platform with: Axiomsys CASE tool (01/01/2000)
  ADSP compatible development board
  ADSP C runtime library
  Virtuoso Operating system
  C language development environment
  Configuration management tool
- Simulators of the other OBS components for the task interfaces testing (output of WP 1.4) (when necessary, starting from 01/06/2000)

WP outputs
- Logical model: the sections relevant to this task (11-08-2000)
- Software specifications: the requirements relevant to this task (11-08-2000)
- Architectural design: the sections relevant to this task (11-08-2000)
- Source code (02-04-2001)
- Executable files (either the single executable or the complete image of the OBS, depending on the operating system implementation) (AVM, PFM Issues)

C.4 Work Package 1.4 – Data packetiser

Objective: to develop the software task necessary to packetise the science/HK/event data according to the ESA telemetry standard and the FIRST packet utilisation standard.

WP description: This work package will be composed by the following activities:
- task design: logical decomposition
- task design: architectural design
- task design: detailed design-pseudocode
- code development: source C code
- software testing: unit tests
- software testing: integration tests

WP inputs:
- FIRST/PLANCK Packet Structure Interface Control Document (01/01/2000)
- OBS URD (DRAFT FORM, 01/01/2000)
- Development platform with: Axiomsys CASE tool (01/01/2000)
  ADSP compatible development board
  ADSP C runtime library
  Virtuoso Operating system
  C language development environment
  Configuration management tool

- Simulators of the other OBS components for the task interfaces testing (output of WP 1.4) (when necessary, starting from 01/06/2000)

WP outputs

- Logical model: the sections relevant to this task (11-08-2000)
- Software specifications: the requirements relevant to this task (11-08-2000)
- Architectural design: the sections relevant to this task (11-08-2000)
- Source code (02-04-2001)
- Executable files (either the single executable or the complete image of the OBS, depending on the operating system implementation) (AVM, PFM Issues)

C.5 Work Package 1.5 – Health Autonomy Functions

Objective: to develop all the software autonomy functions.

WP description: This work package will be composed by the following activities:

- task design: logical decomposition
- task design: architectural design
- task design: detailed design-pseudocode
- code development: source C code
- software testing: unit tests
- software testing: integration tests

WP inputs:

- OBS URD (draft, 01/01/2000)
- Autonomy functions specifications – requirements (?)
- Development platform with: Axiomsys CASE tool (01/01/2000)
  ADSP compatible development board
  ADSP C runtime library
  Virtuoso Operating system
  C language development environment
  Configuration management tool
- Simulators of the other OBS components for the functions interfaces testing (output of WP 1.4) (when necessary, starting from 01/06/2000)

WP outputs

- Logical model: the sections relevant to this task (11-08-2000)
- Software specifications: the requirements relevant to this task (11-08-2000)
- Architectural design: the sections relevant to this task (11-08-2000)
- Source code (07-05-2001)
- Executable files (either the single executable or the complete image of the OBS, depending on the operating system implementation) (AVM, PFM Issues)

C.6 Work Package 1.6 – AVM Issue

Objective: to perform the AVM AIV procedure.

WP description: This work package will be composed by the following activities:

- to perform system integration tests
- to perform the in house acceptance tests with the Spacecraft and Subsystems simulators
- to perform the in house integrated (SW+HW) AVM acceptance tests with the EGSE and the Subsystems simulators
- to perform the AVM acceptance at the consortia premises.
- to provide test reports

WP inputs:
- OBS URD (draft, 01/01/2000)
- DPU/ICU OBS SVVP (draft available from 05/2001)
- Boards from industrial contractor (06/2001)
- Spacecraft I/F simulator (01/05/2001)
- Subsystems simulators (01/07/2001)
- EGSE (01/07/2001)

WP outputs
- DPU/ICU AVM (20/03/2002)
- On site acceptance tests reports (20/03/2002)

C.7 Work Package 1.7 – PFM Issue

Objective: to maintain the OBS until the PFM delivery.

WP description: This work package will be composed by the following activities:

- to update the OBS according to new specifications/requirements
- to execute all the tests necessary to verify and accept the implemented changes.

WP inputs:
- AVM package: OBS code and documentation (20/03/2002)
- new specifications (if any)
- Development platform with: ADSP compatible development board
  ADSP C runtime library
  Virtuoso Operating system
  C language development environment
  Configuration management tool

WP outputs
- Documentation updates (DDD mainly, other documents updates only if explicitly requested)
- Source code
- Executable files (either the single executable or the complete image of the OBS, depending on the operating system implementation)

C.8 Work Package 1.8 – Documentation

Objective: to provide all the documents necessary to support and describe the software development process.

WP description: This work package will be composed by the following activities:

- To write/maintain all the deliverable documents (see section 3) in accordance to the ESA AD2, AD3 standards and to the document management indications provided by the three Consortia.
- To keep the deliverable documentation under configuration control.
- To write the non deliverable documents (technical notes/reports) requested by the Consortia as support to the OBS description/status understanding.

WP inputs: - AD1, AD2, AD3
- All the applicable documents on which will be based the compilation of the user requirements (mainly: all AD4, the FIRST/PLANCK Operations Interface Requirements, Document, the FIRST/PLANCK Packet Structure Interface Control Document, FIRST Operations Scenario Document, the FIRST Instrument Commanding Concepts, all the documents describing the instruments operating modes) (starting from 01/09/1999)
- CASE tool outputs: logical model diagrams, architectural design diagrams, state transitions tables, data dictionary
- Test reports to be included into a OBS STD
- Configuration management tool

WP outputs - OBS URD (draft, 01/01/2000; first issue ISVR, PDR)
- OBS SSD (draft , 01/07/2000; first issue IBDR/IHDR)
- OBS DDD (draft 01/07/2001; first issue ICDR)
- OBS SVVP (draft 01/05/2001; first issue ICDR)

C.9 Work Package 1.9.1 – Support activities: Virtuoso OS

Objective: to install, analyse and evaluate the EONIC Virtuoso Operating System. In particular: to check with dedicated benchmarks the performances of the ADSP with the OS running on it and to analyse the way the OS linker creates the software images.

WP description: This work package will be composed by the following activities:

- participation to information/training courses/seminars on the Virtuoso OS.
- To install the OS on the ADSP PC board and to design and perform benchmarks for the VIRTUOSO evaluation
- To execute dedicated tests on the linker behaviour.

WP inputs: - Development platform with: ADSP compatible development board (01/01/2000)
ADSP C runtime library
C language development environment
Virtuoso Operating system (01/05/2000)

WP outputs: - Final Virtuoso OS evaluation report (01/07/2000)

C.10 Work Package 1.9.2 – Support activities: Test Software

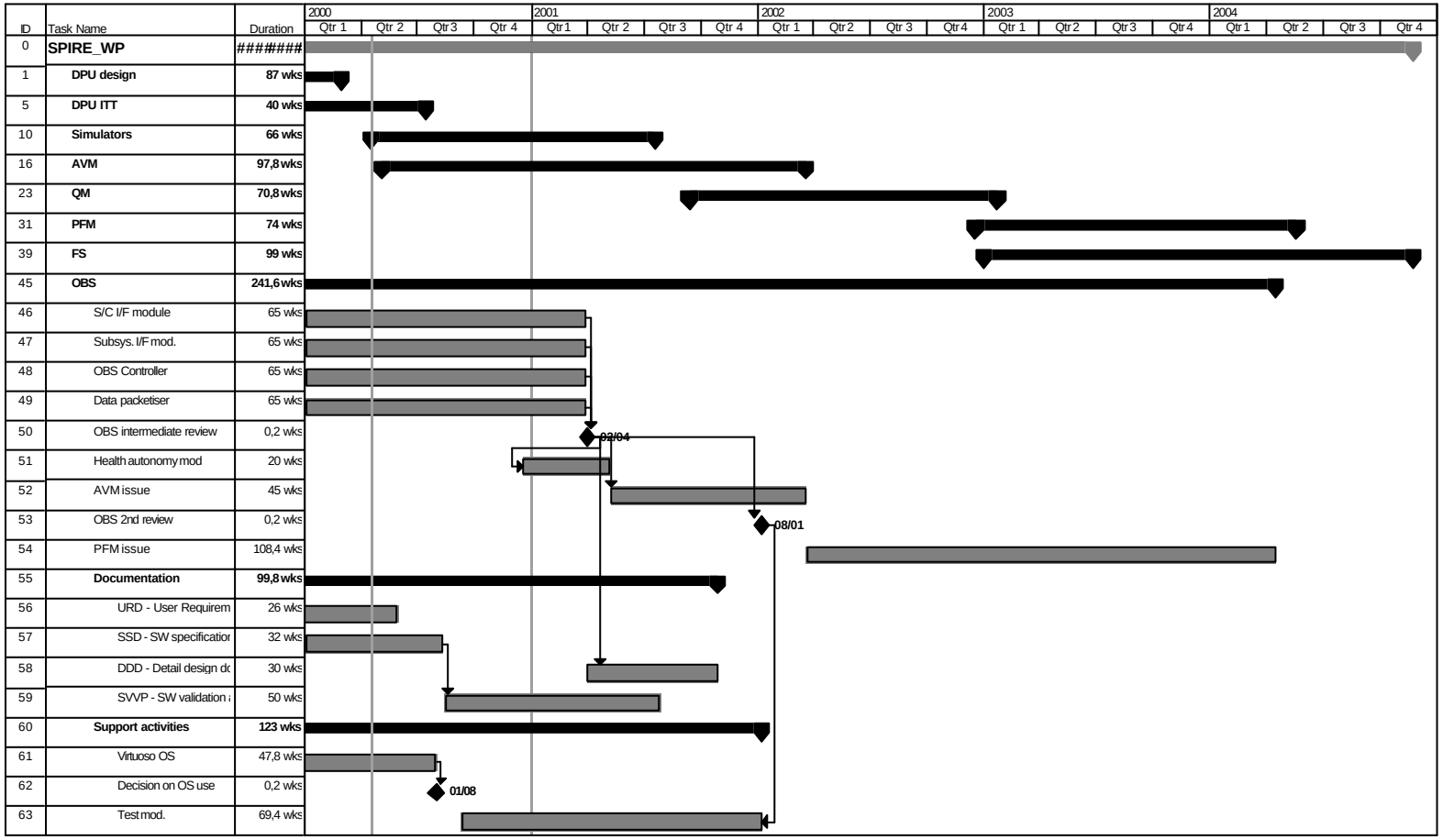
Objective: to develop all the software simulators/stubs necessary to perform the testing activities.

WP description: This work package will be composed by the following activities:

- code development: components stubs, simulators
- to provide test input data and, when possible, test expected results

WP inputs: - DPU/ICU OBS SSD (01/06/2000)
- Development platform with: ADSP compatible development board
ADSP C runtime library
Virtuoso Operating system
C language development environment
Configuration management tool

WP outputs: - test procedures/data to be included into the test reports


ID
Task Name
Duration
2000
2001
2002
2003
2004
Qtr 1
Qtr 2
Qtr 3
Qtr 4
0 SPIRE_WP ########
1 DPU design 87 wks
5 DPU ITT 40 wks
10 Simulators 66 wks
16 AVM 97,8 wks
23 QM 70,8 wks
31 PFM 74 wks
39 FS 99 wks
45 OBS 241,6 wks
46 S/C I/F module 65 wks
47 Subsys. I/F mod. 65 wks
48 OBS Controller 65 wks
49 Data packetiser 65 wks
50 OBS intermediate review 0,2 wks
51 Health autonomy mod 20 wks
52 AVM issue 45 wks
53 OBS 2nd review 0,2 wks
54 PFM issue 108,4 wks
55 Documentation 99,8 wks
56 URD - User Requirem 26 wks
57 SSD - SW specificatior 32 wks
58 DDD - Detail design dc 30 wks
59 SVVP - SW validation ; 50 wks
60 Support activities 123 wks
61 Virtuoso OS 47,8 wks
62 Decision on OS use 0,2 wks
63 Test mod. 69,4 wks
02/04
08/01
01/08