

Generic Requirements Management and Verification Process for Ground Segment and Mission Operations Preparation

Dr. Frank Wallrapp¹ and Andreas Lex²

German Space Operations Center, DLR Oberpfaffenhofen, 82230 Wessling, Germany

A major program at GSOC required the implementation of a requirements management tool. GSOC selected the software package DOORS for that purpose. The positive experience gained in this project, lead to the aim of establishing an in-house standard for requirements management and verification related topics. In order to achieve this goal a number of initiatives have been started:

- An internal training program, designed for the special needs of an operations center, was set up to provide regular training sessions. The training is split between a basic DOORS user training and special sessions on specific topics for key personnel (e.g. project structuring)
- A generic DOORS structure for an exemplary project was developed. This contains templates for all relevant documents and elements. The structure shall be used by all projects as basis for project-specific customization.
- A GSOC standard set of templates was generated to create office documents from DOORS data by using Rational Publishing Engine (e.g. requirements specification, test specification, test report).
- A basic set of DOORS extensions (DXL scripts) is provided to all users. The user gets a set of additional functions (e.g. enhanced import / export, configurable data analysis) without the necessity to develop own scripts.

These initiatives increased the project managers' awareness about the benefit of using a standard requirements management approach. They also improved the acceptance of using DOORS among the project staff. Moreover, the initial effort for new projects to implement the requirements management process using DOORS is significantly reduced.

I. Introduction

In a major project at GSOC a requirement management process was established which was supported by a commercial software tool. The major benefit for this project was simplified tracking of the requirements and their verification as well as the effective reporting for the customer¹.

The positive experience gained in this project was the motivation to establish an in-house standard that covers the needs in mission preparation. This standard is based on the process used in the former project, taking also the lessons learned into account. Within the frame of this standardization initiative various activities were performed as described in this publication.

II. Training Program

For the establishment of this in-house standard it was very important to setup a training program which takes into account the needs of the different user groups. Three major groups can be identified within a project team: the common DOORS-user (e.g. subsystem-engineers), key users within the project (e.g. project manager, quality manager, systems engineer) and the project database administrator.

¹ Project Manager, German Space Operations Center, frank.wallrapp@dlr.de.

² Quality Manager, German Space Operations Center, andreas.lex@dlr.de.

A. Basic DOORS User Training

The basic user training is a one day course in the handling of the tool (IBM Rational DOORS) that is used to support the requirements management process. Additionally it gives an introduction on the generic requirements management process and the master project (see §III). The format of the training is a combination of presentation and practical exercises. The table of content of this course consists of all basic features (e.g. basics of the database, modules, objects, views, links, import, export) which could also be found in commercial courses. The highlight of this course is the practical part which follows our GSOC specific workflow (see §III).

B. Project Structuring

In the beginning of each new project, a project-specific structure and process has to be set up by the project manager that takes into account project-specific boundary conditions. These activities are performed in the form of workshops in which the key persons from the project (e.g. project manager, quality manager and test manager) and the responsible for the master project elaborate a solution. These workshops are repeated during the project to adapt to new boundary conditions.

C. Project Administration in the DOORS Database

Every project needs to assign an administrator who takes care about the project-specific data in DOORS. The duties of this administrator are larger than those of the usual users and therefore not covered by the basic training program. Currently those people are trained on the job by other experienced DOORS project administrators.

III. Master Project

A generic project was set up including all components required for a real project with respect to requirements management. It is based on the experience gained in former GSOC missions that used and that did not use DOORS by conducting workshops with GSOC's project managers. Also ECSS guidelines and DOORS practices were taken into account. This master project shall serve as a customizable template for future satellite mission projects at GSOC. It consists of the components described in the sections A to D of this chapter.

A. Requirements Management and Verification Process during the project Lifecycle

This section describes the process of requirements management and verification from Phase A to Phase D of the master project. An overview of the process is shown in Figure 1.

1. Phase A: Import of Customer Requirements and export of Compliance Matrix

In phase A, the customer inputs are imported into the database. Typically a "Statement of Work (SOW)" and/or a "Customer Requirement Document (CRD)" are delivered (orange boxes in Figure 1). A direct import is possible, if the format of the document is WORD, EXCEL, any proprietary DOORS format or ReqIF. For all requirements, a responsible project member is assigned. The compliance to these customer requirements is stated in separate attributes. Afterwards, the modules are exported and sent to the Customer serving as "Compliance Matrix". If the customer is also using DOORS, the ReqIF format shall be used; otherwise an EXCEL export will be generated.

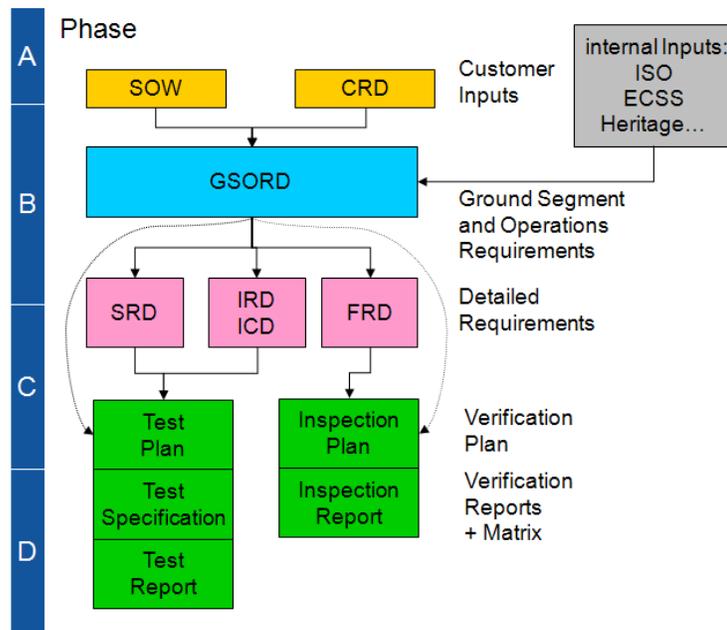


Figure 1. Evolvement of the DOORS database structure during the project phases.

2. Phase B: Derivation of high-level and detailed requirements.

The agreed compliance matrix is the starting point for the phase B activities. The main task in this phase is to build a list of “Ground segment and operations requirements” in the GSORD module which shall form the basis for the project (blue box in Figure 1). This list shall hold all high-level requirements that have to be met in the project. It shall cover customer and internal requirements. To achieve this goal, all customer inputs are reviewed and translated into verifiable requirements for GSOC. By doing so, one or more requirement is written for each customer requirement, where a full or partial compliance was stated. DOORS links are established from the GSORD to the customer requirements (arrows between the orange boxes and the blue box in Figure 1).

Additionally, internal requirements shall be written, to ensure that the project is executed according to internal and normative guidelines, to bring in experience from former missions and to fill gaps in the requirements received from the customer. The ground segment and operations requirements are then classified by filling various attributes like “Requirement Type”, “Applicable System”, “Verification Method” and “Verification Milestone”. A Screenshot of the GSORD module at this stage is shown in Figure 2. The column “Source” is filled automatically by a DXL script (see section C) and shows the identifier of the linked requirements in the SOW module. After an internal review cycle, the ground segment and operations requirements are exported as formal document (see section D1).

After the ground segment and operations requirements were agreed, these high level requirements are refined into more detailed requirements e.g. software-, facilities- and interface requirements (pink boxes in Figure 1) and the attributes filled as described above. DOORS links are established from the detailed requirements to the ground segment and operations requirements (arrows between the blue box and pink boxes in Figure 1). After an internal review cycle the requirements are exported as “Software Requirement Document”, “Facility Requirement Document” and “Interface Requirement Document”

ID	Source	Priority	System	Respor	Level	Milestone	Method	Status	Justification
2.2.3 Mission Data System (MDS)									
2.2.3.1 SCOS2000									
GSRD-0225	SCOS shall be able to command in real-time	-SOW-0043	Mandatory	FOS	SE	CMP	TAR	T	Approved basic functionality
GSRD-0226	SCOS shall be able to command time-tagged	-SOW-0043	Mandatory	FOS	SE	CMP	TAR	T	Approved basic functionality
GSRD-0227	SCOS Dummy Requirement 1	-SOW-0043	Mandatory	FOS	SE	CMP	TAR	T	In Work basic functionality
GSRD-0228	SCOS Dummy Requirement 2	-SOW-0043	Mandatory	FOS	SE	CMP	TAR	T	In Work basic functionality

Figure 2. GSORD in Phase B

3. Phase C: Planning of the Verification Program

In Phase C, the Verification program is planned based on the ground segment and operations requirements (GSORD) and the detailed requirements (SRD, IRD/ICD, FRD). DOORS modules are created for tests and inspections (green boxes in Figure 1). Inside these modules cases for test and inspection are defined and grouped. Within the groups the attributes for “Verification Phase”, “Objectives” and “Constraints” shall be filled. The cases will be defined in more detail in the phase D. At the current stage they have to be linked to the requirements they are supposed to verify. At the end of Phase C every requirement that was classified as ”verified by test” has to be linked to a test case and every requirement that was classified as “verified by inspection” has to be linked to an inspection activity. If a top-level requirement in the GSORD is broken down to detailed requirements, then the verification events shall be linked to the detailed requirements only. Furthermore, it is assumed, that all requirements contained in the FRD are verified by “Inspection”, whereas all requirements contained in the SRD and IRD are verified by “Test”.

Figure 3 shows the test module at this phase of the project. The first row in this example is the Test Group “SCOS2000”. For this group the attributes required for the planning of the tests have been filled. The two rows below represent test cases inside this test group. Requirements from the GSORD are linked to them. The IDs of these requirements are shown automatically in column “Reqs” by using a DXL Script. The inspection module looks similar.

After the test and inspection modules are filled, the “Verification Plan” document is created from them (see section D).

	Number	Reqs	Objective	Constraints	Milestone	System	Level
1.1.3.1 SCOS2000	T-FOS-CMP-1.3.1		These Tests shall verify that SCOS2000 satisfies all requirements and is ready to be integrated into the operational system	<ul style="list-style-type: none"> • Simulator is connected • Flight Database is available in preliminary version 	TAR	FOS	CMP
Commanding Test Case	T-FOS-CMP-1.3.1-1	GSRD-0225; GSRD-0226			TAR	FOS	CMP
Dummy Test Case	T-FOS-CMP-1.3.1-2	GSRD-0227; GSRD-0228			TAR	FOS	CMP

Figure 3. Test Module in Phase C

4. Phase D: Execution of the Verification Program

After integration of the ground system the verification can take place. With respect to inspections, the inspections module is expanded by attributes that detail the inspection activity like “Expected Finding”, “Location”, “Result”, “Date”, and “Person Inspecting” Afterwards the inspections themselves are performed and the attributes filled. The filled module is finally exported as “Inspection Report”.

With respect to tests, they are performed in two stages. First, roughly two weeks before the test a test specification is generated. For that reason various attributes defining the test in detail like “Configuration”, “Pass Fail Criteria”, “Test Steps” and “Test Scope” are added to the module and filled. A subset of the test module which includes the tests to be performed in the next session is exported as “Test Specification”. Second, the test is performed as defined in the test specification. In order to track the test results, various attributes are added to the test module and filled like “Date”, “Result”, “Engineer”, “As-Run Procedure” The subset of test cases that was performed in the session is then exported as “Test Report”.

After all inspection and test activities are completed “Verification Matrices” can be exported which contain the traceability from the requirements to the verification events (see section D2). Since all corresponding elements inside the DOORS database are linked with each other, matrices between the verification and all levels of requirements (customer, high-level, detailed) are possible. These matrices are the formal prove that all requirements, either internal or set forth by the customer, are met and are the basis for decision that the ground segment is green for launch.

B. DOORS Database Structure

This section describes the structure of the master project in the DOORS database.

1. Folder and Module Structure

The organization of folders (yellow boxes) and DOORS modules (blue and green boxes) is shown in Figure 4.

On the top level, the project is divided into the folders ”10 - customer input”, ”20 - technical requirements specification” and ”30 - verification”. Folder 10 contains all documents which hold requirements set forth by the customer. They are received in the two documents SOW and CRD. These are the input for the whole process. Folder 20 contains all internal requirements or those that are derived from the customer requirements. There are two levels of derived requirements (compare Figure 2). The top level requirements are contained in the GSORD folder and the detailed requirements in the SRD, FRD and IRD folders. In each folder one DOORS module contains the requirements

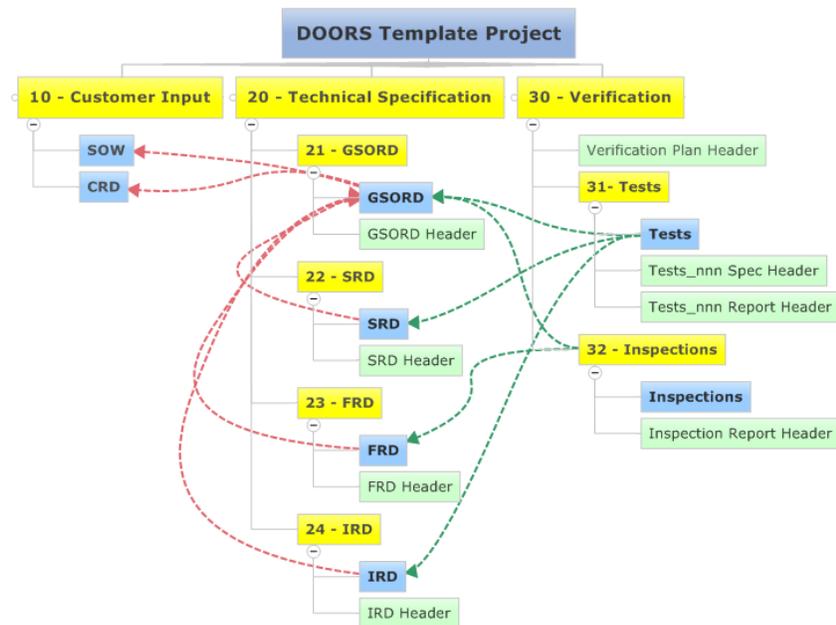


Figure 4. Complete DOORS database structure

themselves (blue boxes) and one DOORS module holds all other information necessary to produce the respective document (green boxes) (see chapter D). These modules are named "... Header" and contain e.g. the title page, introductory chapters and all other content, which completes the document. Folder 30 contains the verification activities with respect to tests and inspections. There exists one module for all test cases and one module for all inspection cases (in blue). Also in this area, there are supporting modules, which are required to generate documents based on the data (green boxes) (see chapter D1). For the tests, various test specifications and reports may be exported, whereas for inspections one single inspection report is foreseen (see section 4 of previous chapter).

2. Link Structure

The master project contains two types of links, each of them contained in a separate link module. Red arrows in Figure 4 indicate links of type "satisfies". These links start on the derived requirements and end on the source requirements. They are used for linking detailed requirements (SRD, FRD, IRD) to the top level requirements (GSORD) and for linking top level requirements (GSORD) to customer requirements (SOW, CRD). Green arrows in Figure 2 indicate links of type "verifies". These links start on verification events, e.g. test and inspection cases, and end on top level or detailed requirements. In principle, test and inspection cases could be linked to requirements contained in any requirement module. However, in Figure 4 the assumption is made that in the SRD and IRD the requirements are verified by tests and in the FRD the requirements are verified by inspections. Only the GSORD holds requirements that are verified by test and requirements that are verified by inspection. With this link structure complete traceability from the customer requirement via the derived top level and detailed requirements to the verification events is established.

3. Structure within the modules

The table of contents within all data containing modules (derived requirements and verification events) is based on the same structure, which in turn is based on a generic product tree that was also developed in the frame of GSOC's standardization activities. The high-level structure of the result is shown in Figure 5. The complete tree is used for structuring the GSORD module, since the GSORD is supposed to contain requirements that cover all project areas. The other modules contain a subset of this structure. For example, the SRD contains only software products and the test module contains only those products that are foreseen to be tested. This approach ensures that all modules have a similar structure.

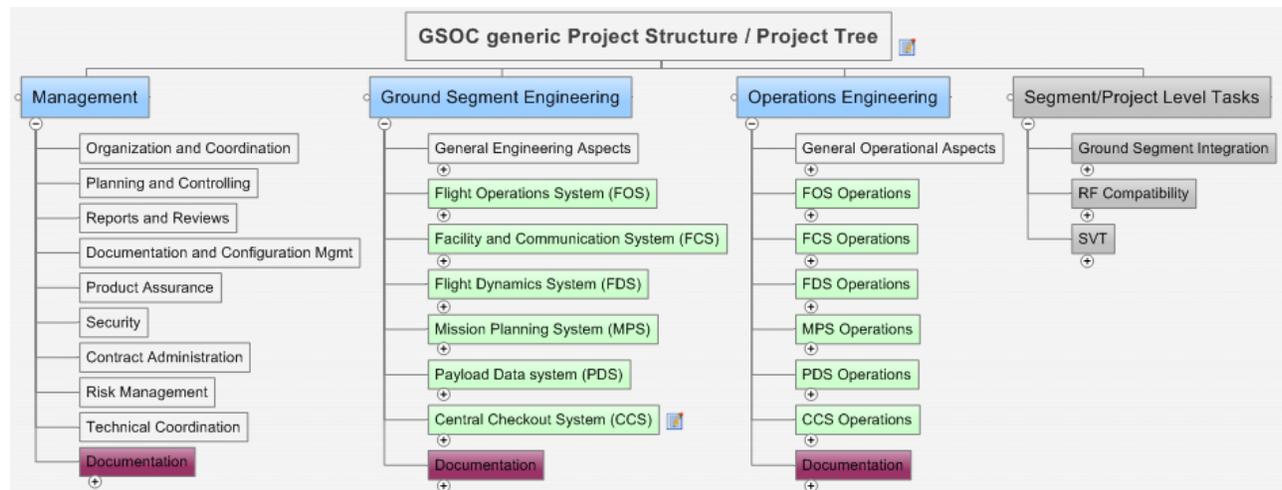


Figure 5. Base structure for all modules

4. Attributes and Attribute Types

After the required modules have been set up, the attributes and the types of these attributes have been defined for the whole project. This was performed in a large table which holds a list of all attributes contained in the project. For each attribute, the following information is given:

- Modules in which the attribute is used.
- Type of the attribute (e.g. text, date, Boolean, or user defined)
- Indicator if the attribute value is calculated by a script (DXL Attribute), see section C.
- For DXL Attributes: Source Code of the script

- Description of the attribute
- Scope of the attribute (on module level or object level)
- For user defined types: Possible attribute values.

A part of the table is shown in Figure 6. In total this table contains 75 attributes. By applying this table, it is ensured that the naming of the attributes and the possible values are consistent throughout all modules. Furthermore, this approach simplifies the creation of standardized views (see next chapter).

	A	B	C	D	E	F	G	H	I	J	L	
1	Gen Info.			used in Module								
2	Attribute Name	Type	DXL	Type	CRD SOW	GSRD	xRD	Tests	Insp.	Header	Description	DXL Code
30	G Req Level	E Level		O		X	X				Level at which the requirement is specified	
31	G Req Priority	E Priority		O		X	X				Priority of the Requirement	
32	G Req Responsible	E Responsible		O	X	X	X				Responsible for the fulfilment of the requirement	
33	G Req Source	Text	X	O		X	X				(DXL) Source for the requirement; calculated by DXL using links	#include //ocs.ocs.office/p_DOORS/_active OUTLINKS = true OUTLINKMOD = "satisfies" ATTR[0] = "G_Req_ID"
34	G Req Status	E Req Status		O		X	X				Status of the requirement	#include //ocs.ocs.office/p_DOORS/
35	G Req System	E System		O		X	X				System for which the requirement applies	
36	G Req Type	E Type		O		X	X				Type of the requirement	

Figure 6. Attribute Plan

5. Views

Pre-defined views were created for every module. The names of the views imply their use e.g. “Requirement Classification” and “Verification Matrix”. The views show only the subset of attributes that are required for a given project stage or task. This measure supports establishing a structured workflow and to help the users to see and work with the information that is currently required.

C. Extension Scripts

DOORS features a powerful scripting language (DXL), which can be used to analyze data or extend the functionalities of the program. Two types of scripts can be discriminated: Public available scripts and own scripts made specifically for use in the master project. Concerning publicly available scripts, the internet was searched and those scripts that were judged to be of interest for all users were added to the standard installation procedure. They are ready to use for every user afterwards. Examples are enhanced import and export to EXCEL, enhanced history function, a function to copy views to other modules and many more.

With respect to own scripts, these are all so-called DXL attributes, which means, that attribute values are calculated by using the script. Usually, the script has to be put into the attribute definition for every attribute in every module that uses the script. This is very inefficient, since often the same or slightly different scripts are used in various attributes spread across various modules. Therefore, enhancements of the scripts or fixing bugs would be very time-consuming. In order to overcome this problem, there exists only one instance of each script for the whole database, which is located on a central network drive, to which every user has read-only access. The attribute definition itself contains only a command to include the script file from the central network drive and some definitions of variables to configure the script for this specific attribute. Moreover, to keep the number of scripts that have to be maintained low, the philosophy is to have one script, which can have various applications and can be configured for a specific one by setting the variable values in the attribute definition. A good example is the “Harvest Data Script” which collects data from various sources, formats and sorts the inputs and puts the result in the attribute value. The same script can be used to collect attribute values from linked objects from other modules, from children of the current object, from children of the linked objects or from a combination of all. The number of attributes that can be evaluated from the source is not limited. This script is largely used whenever information from other linked modules must be shown like e.g. in the column “Source” in Figure 2 and “Reqs” in Figure 3. This feature is the basis for all types of traceability and verification matrices (see section D2)

D. Standard set of exporting templates

For other parties involved in the project or to comply with contractual obligations, formal documents need to be generated from the data inside the database. There are two types of exports: WORD exports and EXCEL exports. Both of them are handled differently and are applied for different documents.

1. WORD Exports

Documents which do not require large tables to adequately display the DOORS data are exported in WORD format, e.g. requirements documents and verification documents. Usually a document is generated by combining a header module and a data module (see Figure 4). By using the link functionality and the “Harvest Data” script (see chapter C), also data contained in linked modules is contained in the data module and thus can be exported...

The tool used for exporting these documents is “IBM Rational Publishing Engine (RPE)”. The process is shown in Figure 7. RPE is a very complex tool which allows generation of a document from various sources inside the DOORS database. In order to use the tool, templates in RPE and WORD have to be created for every document type. Whereas one WORD template can be used for all documents, a specific RPE template needs to be created for every document listed in Table 1. The templates are built to work with the DOORS modules from the master project. The idea behind generation of the documents is that there is no need to manually edit the WORD file after exporting. For that reason, all content that is not contained in the module containing the data itself is written in so-called “Header Modules” in DOORS (see section B1).

This module contains the title page, information for header and footer, additional introductory chapters and every other content required to complete the “formal” document. In the exporting process, the information contained in the header and data modules is combined in a single document, which is ready to be printed or archived without further manual editing.

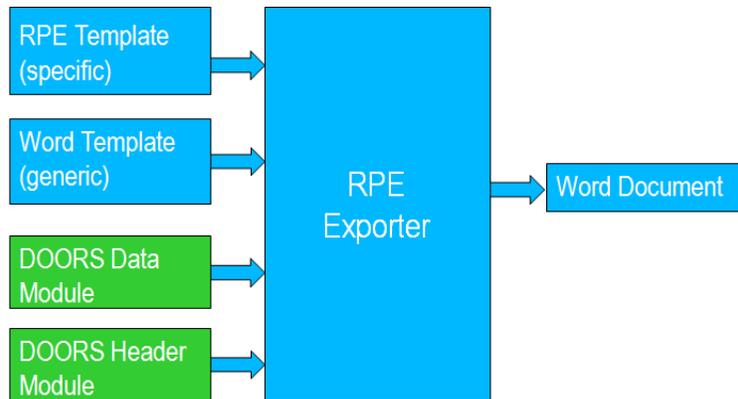


Figure 7. WORD exporting process

This module contains the title page, information for header and footer, additional introductory chapters and every other content required to complete the “formal” document. In the exporting process, the information contained in the header and data modules is combined in a single document, which is ready to be printed or archived without further manual editing.

Figure 8 shows some examples of WORD exports. Figure 8a shows the representation of a requirement as contained in the GSORD in a WORD Export. It is one of the requirements that can be seen in its DOORS representation in Figure 2. The WORD exporter allows totally flexible configuration of the output. In this case one requirement and all of its attributes are put together in one WORD table. Please note that in this example also information from other modules is shown. The entry in the field “Source” is collected by following links to the customer SOW and shows the requirement ID of the customer requirement from which the shown requirement is derived. Figure 8b shows an excerpt of the “Verification Plan” document. It shows the chapter that describes the Test Group named “T-FOS-CMP-1.3.1”. The DOORS representation of this test group as contained in the “tests” module can be seen in Figure 3. In this example, the attributes are put below subheadings in this chapter. Also here, information from other modules is contained. The Requirements IDs from the GSORD module of the requirements that are verified by all tests cases belonging to the test group are displayed (Numbers starting with “GSRD-”).

GSRD-0226	Priority: Mandatory	Status: Approved	Resp.: SE
Date: 2012-04-19	Level: CMP	Ver. Method: T	Ver. Milestone: TAR
System: FOS	Type: FUN	Internal: False	
<i>SCOS shall be able to command time-tagged</i>			
Source:	-SOW-0043		
Refined in:			
Information:			
Justification:	basic functionality		

Figure 8a. Example of a requirement after WORD Export of GSORD

<p>5.1.1.3.1 <i>SCOS2000 (T-FOS-CMP-1.3.1)</i></p> <p>OBJECTIVE OF THE TEST</p> <p>These Tests shall verify that SCOS2000 satisfies all requirements and is ready to be integrated into the operational system</p> <p>CONSTRAINTS TO BE CONSIDERED</p> <ul style="list-style-type: none"> • Simulator is connected • Flight Database is available in preliminary version <p>TEST SCHEDULE</p> <p>Test to be performed until TAR</p> <p>REQUIREMENTS TO BE VERIFIED</p> <p>GSRD-0225; GSRD-0226; GSRD-0227; GSRD-0228</p> <p>TEST CASES</p> <p># T-FOS-CMP-1.3.1-1: Commanding Test Case</p> <p># T-FOS-CMP-1.3.1-2: Dummy Test Case</p>

Figure 8b. Example of Test group after WORD Export of Verification Plan

2. EXCEL Exports

Data, which needs to be displayed in large tables in order to be displayed usefully, is exported in EXCEL format. This EXCEL table is annexed to a WORD document which holds all the other information to constitute a “formal document”. Documents that are generated in this format are usually matrices based on one module, which contains also linked information from other modules. An example is shown in Figure 9. A screenshot of the GSORD module is shown in a) and the corresponding EXCEL export in b). The last column shows the status of the test cases that are linked to the requirements. Such a representation may serve as a verification matrix for the GSORD document giving evidence that all requirements are verified.

Req_ID	Internal?	Milestone	Met	System	Respor	Level	Verification_Reference
2.2.3 Mission Data System (MDS)							
2.2.3.1 SCOS2000							
GSRD-0225	False	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-1 Pass 19 April 2012
GSRD-0226	False	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-1 Pass 19 April 2012
GSRD-0227	False	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-2 Fail 19 April 2012
GSRD-0228	False	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-2 Fail 19 April 2012

Figure 9a. GSORD module with verification references from Test module (last column)

A	B	C	D	E	F	G	H	I
Req_ID		Internal?	Milestone	Method	System	Responsible	Level	Verification_Reference
2.2.3.1 SCOS2000								
GSRD-0225	SCOS shall be able to command in real-time	FALSE	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-1 Pass 19 April 2012
GSRD-0226	SCOS shall be able to command time-tagged	FALSE	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-1 Pass 19 April 2012
GSRD-0227	SCOS Dummy Requirement 1	FALSE	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-2 Fail 19 April 2012
GSRD-0228	SCOS Dummy Requirement 2	FALSE	TAR	T	FOS	SE	CMP	-T-FOS-CMP-1.3.1-2 Fail 19 April 2012

Figure 9b. EXCEL Export of GSORD module module with verification references from Test module (last column)

IV. Conclusion

The requirements management process setup in a prototype project was a great benefit in mission preparation¹. In order to promote this positive experience various activities were performed to establish an in-house requirements management standard. The initiatives described in this publication helped promoting the benefit of a structured requirements management process. This includes the project managers who have to implement such a process and all other project staff who own a role in this process. The availability of a ready-to-use template project which is accompanied by a process description reduces the effort for establishing such a structure in a new project. Another benefit which leads to a cost reduction is that the structure and process is similar or even identical in different projects. This increases the effectiveness of people working in various projects.

Appendix A: Acronym List

CRD	Customer Requirement Document
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DXL	DOORS Extension Language
ECSS	European Cooperation for Space Standardization
FRD	Facility Requirements Document
GSORD	Ground Segment Requirement Document
IRD	Interface Requirements Document
RPE	Rational Publishing Engine
SOW	Statement of Work
SRD	Software Requirements Document
VP	Verification Plan

Appendix B: Glossary

DOORS	<i>Commercial software tool used for Requirements Engineering and Tracking</i>
Module	<i>Collection of data inside DOORS. One Project inside the database contains various modules. The module has attributes. One module contains various hierarchically structured objects.</i>
Object	<i>Data element inside a DOORS module. One object may represent e.g. a requirement or a test case. Objects can have various attributes.</i>
Attributes	<i>Property of a module or an object. Modules and objects can have an unlimited number of attributes</i>
Attribute Type	<i>The type of an attribute. Base types are e.g. Boolean and Text. Custom Attribute Types can be generated which consist of a list of allowed values.</i>
ReqIF	<i>XML file format that can be used to exchange requirements, along with its associated metadata, between software tools from different vendors. The ReqIF format also defines a workflow for transmitting the status of requirements between partners</i>
View	<i>Subset of data contained in a DOORS module. A view is defined by applying filters to the data and show only selected attribute.</i>

References

¹Wallrapp, F and Lex, A, “Requirements management for mission preparation at the German Space Operations Center (GSOC)”. *SpaceOps Conference*, 2010-2010, 2010