Quality Assurance Plan D9.1.1

––

Deliverable Number: D9.1.1
Contractual Date of Delivery: month 3
Actual Date of Delivery: 27/07/2001
Title of Deliverable: Quality Assurance Plan
Work-Package contributing to the Deliverable: WP9
Type of the Deliverable: INT*
Nature of the Deliverable: R**
Author(s): Umberto Straccia (CNR)
straccia@iei.pi.cnr.it
Version: First version (20/4/2001)

Abstract: This deliverable defines the software quality assurance plan to be applied to the software products developed by the Consortium partners as part of the CYCLADES project. Is also specifies standards and procedures to be developed for the management of the project along with a set of uniform rules for coding any document which will be issued in relation to the CYCLADES project.

Keyword List: Project; Management; Quality; Procedures.

Table of Content

Introduction

The main goal of a quality assurance plan is to identify the objectives and the expected outputs and to develop criteria for the validation of the outputs.

This deliverable defines:

- The Software Product Quality Assurance plan to be applied to the software products developed by the Consortium partners as part of the CYCLADES project.

  Its purpose is to describe the procedures and the rules to be applied and the tools to be used in order to ensure the quality during the design, development, testing and validation phases of the project.

- Standards and procedures to be developed for the management of the CYCLADES project. The approach taken is to concentrate on three areas:

  - The first one is concerned with the exchange of information between the Work package groups, with respect to the specification of the products which are deliverable from one Work package to the others.
  - The second is concerned with key Quality Processes to be conducted by all Work packages, to which they are applicable.
  - The third is concerned with procedures deemed necessary for proper control of the quality of the various Work packages and of the management procedures.

- A set of uniform rules for coding any document which will be issued in relation to the CYCLADES project.

It is recognized that each of the organizations involved in CYCLADES has its own procedures. The extent of this Quality Assurance Plan is confined to the procedures necessary to facilitate the ready flow of information and quality control throughout the project, by applying a reasonable level of commonness, content, and process.

Procedures other than those listed may be found necessary at a later stage. Similarly, it may be felt, at a later stage, that not all of the standards or procedures are applicable to this project.

1.1 Scope of the Software Product Quality Assurance Plan

The Software Product Quality Assurance Plan shall apply from the software specification phase until the end of the validation phase.
The different items to be handled under the full configuration tools are:

- All deliverable documents and available documents.

- All software products implemented by the partners.

These rules will be followed by all partner teams involved in the CYCLADES development.

1.2 Project characteristics

There are three main streams of work in CYCLADES:

- The study of the user requirements and the system specification
- The design and development of a reusable software.
- Experiments to be conducted in a restricted number of field tests to check the validity of the project approaches.

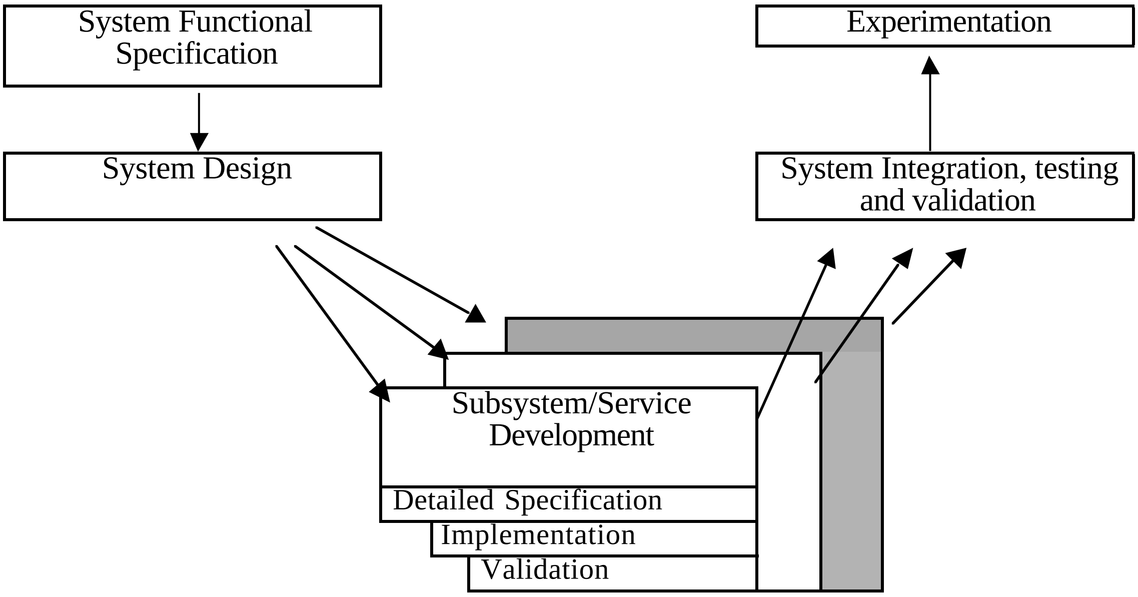
2. Software Development Steps

2.1 Product Life Cycle

The software product may be considered as a system composed of software subsystems/components implemented on different environments.

Therefore the product life is composed of the definition of the requirements and the design of the system, the independent development of each subsystem/component, followed by the integration of the subsystems and the delivery of the product to the users for the acceptance-testing phase.

The following figure illustrates the different phases of the product life cycle and their sequencing.


System Functional Specification
Experimentation
System Design
System Integration, testing and validation
Subsystem/Service Development
Detailed Specification
Implementation
Validation

A critical phase is the system integration phase, since it includes the integration of several subsystems/components. The duration of this phase requires that it starts very early in the project life cycle, however, this phase is conditioned by the subsystem/component deliveries. Thus, the sooner the subsystems/components are delivered to the integration team, the sooner can this phase be successfully completed.

The solution retained to meet this need is to start subsystem/component development before the end of the system specification and definition phases. This is possible thanks to the large experience of the partners in this area and to an accurate knowledge of the subsystem/component functionalities.

2.2 Software Development Phase Description

2.2.1 System Functional Specification Phase

This phase is the first phase of the software life cycle. The goal of this phase is to answer the question: “What function is to be realised by the software system?” (“regardless” of the implementation, to focus on the function).

The User Requirements Report (URR) collects user needs for the creation of a software infrastructure to support scholars both individually and as members of networked communities when interacting with large interdisciplinary electronic (e-print) archives. After agreement, the User Requirement Report (URR) must be taken into account by the configuration management. Any requirement change leads to report revision.

This phase prepares the further phases by defining the organization of the project management and the Acceptance Test Strategy.

Input document

- User Requirement Report.

Output document

- Quality Control Plan.
- Project Management Plan.
- CYCLADES Global System Specification Report (SSR).

The URR is completed by the CYCLADES SSR, which is a technical document that translates the user requirements into system functionalities, aggregates these into functional architectural components, and defines their interfaces and global functional system architecture.

2.2.2 System Design Phase

The system design consists of the following steps:

- Defining the operational architecture, e.g., a service based architecture.
- Detailing the chosen solution i.e., decomposition into subsystems/services.
- Describing the interfaces between the subsystems/services.
- Defining the standards to be adopted.

The preparation of the further phases includes:

- The definition of the integration strategy and the necessary means.
- The description of the sequencing of the development activities.
- The installation of the software environment for the development and for the configuration management.

Input document

Output reports from the previous phase and particularly:

- CYCLADES Global System Specification Report (SSR).

Output document

- CYCLADES System Operational Architecture Report (OAR).
- Integration and Validation Plan.
- Project Development Plan.

2.2.3 Subsystem/Service Development Phases

The development procedure for each software subsystem/service is a process constituted of a set of activities allowing the functional and architectural specifications to be translated into subsystem/service detailed specifications, the detailed specifications to be written in code and the code to be tested, documented and validated for further software system integration.
The figure is a summary of these phases with the main tasks and functional responsibilities:

- Software development.
- Configuration management.
- Software quality control.

Specification				Validation
	Design		Integration	
		Coding		

SOFTWARE DEVELOPMENT of SUBSYSTEMS/SERVICES

Translate	Realize	Write	Integrate	Install
User requirements into: - functions - performance - interface Specification Subsystem Write Subsystem Functional specification	Detailed design Write - Operational architecture report - Detailed specification report - Test plan document	Software modules Test Of software modules	Software modules Execute Integration test Define Installation procedure	Software subsystem Issue Delivery note

CONFIGURATION MANAGEMENT

	Define Nomenclature			Record Items

QUALITY CONTROL

Check Output documentation	Organize Software design review		Check Test planning and test activities	Organize Validation review Control Configuration management activities

2.2.3.1 *Subsystem/Service Detailed Specification Phase*

For each subsystem/service identified in the previous phase, the detailed specification consists in:

1. Specifying the functionalities
2. Specifying the information/data flow
3. Specifying the internal architecture
4. Specifying the interfaces with the other components/services

The preparation of the further phases includes the definition of modular tests.

Input documents

Output documents from the previous phases and particularly:

- CYCLADES System Operational Architecture Report (OAR).

Output documents

- Subsystem/Service Detailed Specification Report (DSR).
- Subsystem/service test plan defining the module and integration tests of the subsystem/service. Taking into account that different subsystems/services vary extremely concerning their complexity and importance for the complete software system it's legal to allow also a variation of test plans concerning the degree of detail and content. So the creation of subsystem/services test plans as a base for successive tests is not necessarily obligatory – it depends on the individual subsystem/service. The decision about that is made by the team leader responsible for the subsystem/service.

2.2.3.2 Subsystem/Service Implementation Phase

For each subsystem/service described in the DSR, the corresponding implementation phase consists in:

- Producing code and pseudo-code if needed.
- Building an executable module with eventual options.
- Performing all the modular and integration tests according to the underlying subsystem/service test plan
- Checking the results of tests to ensure good behaviour in regular and exception cases.
- Defining an installation procedure for the subsystem/service.

Input documents

Output documents from the previous phases and particularly:

- Subsystem/Service Detailed Specification Report (DSR).
- Subsystem/Service test plan.

Output documents and products

- Source code listing, executable and generation procedures.
- Subsystem/Service test document: test plan completed by modular test reports and module integration test reports.

2.2.3.3 Subsystem/Service Validation Phase

The validation consists of proving that the subsystem/service meets the detailed specification.

Input documents

Output documents and products from the previous phases.

Output documents and products

Updated output documents and products from the previous phases:

- Operational Architecture Report (OAR).
- Subsystem/Service Detailed Specification Report (DSR).
- Subsystem/Service test document.
- Source code listing, executable and generation procedures.
- Description of the delivery including the restriction list.

This phase is completed by the subsystem/service validation review, which checks if the subsystem software may be delivered to the team in charge of the system integration.

At the end of this phase, all source code, executable modules and generation procedures are under configuration management.

2.2.4 System Integration Phase

The system integration consists in:

- Building step by step the system according to the integration strategy that has been defined during the system design phase. This is implemented by successive integration of the different validated software subsystems/services.
- Testing the logical system functions against the Detailed Specification Reports
- Defining the system installation procedures.

Input documents

- Software of previous phases and particularly the subsystem object.
- Output documents of the system specification and design phases.

Output documents and products

- System Integration and Validation plan including the result of the test integration.
- Installation guide.

This phase ends by the authorization of the integration team leader for the delivery of the software to the users.

2.2.5 Experimentation Phase

The experimentation consists in:

- To establish a demonstration environment;
- to define a demonstration methodology; to measure user satisfaction, on a large scale, by making full system capabilities available to the user;
- to collect and analyse the evaluation results.

3. Configuration Management Procedures

3.1 Purpose of Software Configuration Management

The Software Configuration Management (SCM) activities help in making changes to software work products an efficient process. Change of work products is necessary throughout their life cycle. Some control over change is necessary:

- To preserve work products integrity:
    - Several conflicting changes can be made concurrently.
    - The correctness of a change needs to be checked to prevent the introduction of defects.
    - The impact of a change on other work products needs to be evaluated.
- To inform impacted people of the change and maintain agreement on shared objects.

One of the ultimate goals of the SCM activities is to ensure the quality of the products delivered to the customer.

This document lists the activities that need to be performed to conform to the SCM activities. Each partner will implement the activities described here, with possible customisation taking into account already existing practices and procedures.

3.2 Glossary

Work Product Any output created by a project. This can include source code of the created programs, associated documentation, analysis and design documents, but also research papers, articles, dictionaries and so on.
A more formal definition is : “Any artefact created as part of defining, maintaining, or using a software process, including process descriptions, plans, procedures, computer programs, and associated documentation, which may or may not be intended for delivery to a customer or end user”.

Configuration item A configuration item is a work product produced by the project and subject to software configuration management activities.

Baseline A set of configuration items on which formal change procedure applies. A baseline is a stable entity, which has been formally identified, reviewed by both “suppliers” and “customers” (see section 6.1 for definitions). Typically, a baseline will be created for the release 1.2 of the *Foo* software product, or the version of an article submitted to a conference call.

3.3 Roles

Each Partner is responsible for implementing configuration management activities for the work product it delivers.

The Partner in charge of coordinating a particular subsystem/service is responsible for coordinating configuration management information and implementing configuration management activities for the joint deliveries of the subsystem/service, e.g., paper delivery.

The Project Director is responsible for implementing configuration management activities for other deliverables such as Annual Reports, Peer Review documents, etc.

The project Director is responsible for reviewing with each Partner that Configuration Management is implemented in a compliant way. He reports possible issues to the Project Management Committee.

3.4 Activities

All the work products that are part of the deliveries (final deliveries or partial deliveries to partners) are subject to the SCM activities. The tools used to build or exploit the work product might also be candidates for being configuration items if their future physical availability is not strongly ensured.

3.4.1 Identification of the software configuration items

A unique Id, a type, and a description are associated with each configuration item. The level of granularity of the configuration items is determined by each Partner according to its need and already implemented practices. One Partner might need to identify each source file as a configuration item whereas another Partner will consider its whole delivery as being a single configuration unit: this can be the case, for example, if the Partner is providing an executable version of a software package that is part of the Background information.

3.4.2 Definition of a baseline

At certain points (usually near milestones), the Partner in charge of a deliverable defines a baseline and asks the Project Director to review it. A baseline lists the configuration items Id and versions of the configuration items that compose it. Major changes to the configuration items that compose the baseline should be reported to the Project Director.

3.4.3 Version control

The software configuration items, base lined or not, are managed through the use of a version control tool that provides:

- Access to previous versions.
- A history and description of the changes made on those items.

The main tool recommended to all partners is CVS, but Partners already using other tools as part of their current practices can continue doing so. Using such a SCM tool, each Partner maintains the configuration management library composed of all configuration items. Each time a change is made to a configuration item a description of that change is entered using the tool facility.

4 Document Standards and Guidelines

The objective of this task is to formalize a set of uniform rules for coding any document which will be issued in relation to the CYCLADES project. These rules are meant to standardize not only the scheduled deliverables which will be delivered to the Commission at the end of each Work Package/Task, but also any other document exchanged between the partners.

Standardization implies fixing a single and clear filing code for all the documents, giving guidelines for a logical structure of the filing code, sketching the forms which will be used for the on-the-field analysis or interviews, etc.

These rules can be modified during the project life as may be necessary.

4.1 Document Identification

Each page of a deliverable starts with a header. The leftmost position of the header contains:

The project number

(Ex. IST-2000-25456)

The filing code for document recording of the deliverable

(Ex. Deliverable D6.1.1)

The rightmost position of the header contains:

The title of the deliverable

(Ex. System testing report)

Each page of a deliverable indicates the page number and the total number of pages at the bottom part in the rightmost position

(Ex. Page 1 of 20)

The bottom part of each page also includes, in the leftmost position

The acronyms of the partners who have participated in the production of the document

(Ex. Originator: CNR, UniDo)

Each page of a document indicates the date of issue of the deliverable, at the bottom of each page in the centre position

(Ex. 07/30/98)

4.1.1 Standard Filing Code

The formal representation of the filing code for the Commission deliverables is:
D1.2.3

For each work package of the contract a given number of deliverables must be produced. Each work package, in turn, is divided into a given number of tasks (one or more) which are sequentially numbered (T1.1, T1.2, etc.). For each task one or more deliverables must be produced.
The meaning of the full code specification is the following:

(D1) The work package number to which the Deliverable corresponds

(2) The task number within each work package to which the deliverable corresponds

(3) The sequence number of the deliverable within each task of each work package

4.2 Formal Rules for Writing

Formal rules describe how a document, and its contents, must be logically organized.

Two classes of documents are considered:

1. Deliverables
2. Documents exchanged between partners

All documents must be written in English.

4.2.1 Deliverables

Each document thus starts with a cover page which contains all the information needed for its identification and retrieval.

4.2.1.1 Cover Page Structure

PROJECT NUMBER

The project number of the CYCLADES contract
(IST-2000-25456)

PROJECT TITLE

The title of the project, both acronym and full title
(CYCLADES – An Open Collaborative Virtual Archive Environment)

DELIVERABLE TYPE

PU-public, Int- Internal within project, Rest-Restricted, IST-Circulation within IST, FP5-Circulation within Framework Programme partecipants

DELIVERABLE NUMBER

The filing code for document recording of the Commission deliverables

CONTRACTUAL DATE OF DELIVERY

The date at which the deliverable was to be consigned to the Commission, as written up in the contract

ACTUAL DATE OF DELIVERY

The date at which the deliverable was actually sent to the Commission

TITLE OF DELIVERABLE

The title of the deliverable, as written up in the contract

WORK-PACKAGE CONTRIBUTING TO THE DELIVERABLE

The work packages that contributed to the production of the deliverable

NATURE OF THE DELIVERABLE

P-Prototype, R-Report

AUTHORS

Author(s) and partner acronym(s) of the persons contributing to the actual writing up of the deliverable

VERSION

Final, first, second, etc.

Date of the last revision (issue date if no revision has been made)

ABSTRACT

A few lines describing the content of the document

4.2.1.2 Second Page

The second page contains the table of contents.

4.2.1.3 Third page

The third page contains an executive summary, outlining the aim, procedure and results of the deliverable.

4.2.1.4 Revision

Revision occurs when updating a part of a document already distributed. A revision results in a change of the VERSION, written up on the cover page.

4.2.2 Documents Exchanged between Partners

Documents exchanged between partners must contain, on the first page:

WORKPACKAGE NUMBER and/or DELIVERABLE NUMBER

The work package and/or the deliverable number for which the document is produced

DATE OF ISSUE

The date at which the document was first issued

AUTHORS

Author(s) and partner acronym(s) of the person(s) producing the document

VERSION

Final, first, second, etc.

Date of the last revision (issue date if no revision has been made)

4.2.2.1 Revision

Revision occurs when updating a part of a document already distributed. A revision results in a change of the VERSION.

4.3 Other recommendations

4.3.1 Textual content

Text must be written in English.

4.3.2 Setting out

All documents must be set out on A4 size paper using single spacing between lines.

4.3.3 Margin

Left and right justified margins as in this document.

4.3.4 Table of Contents

For large documents, the page following the cover page must contain the table of contents of the document.

4.3.5 Minutes of meetings

Minutes of meetings should contain at least the following sections:

- Participants,
- Agenda,
- Inventory of released documents since the last meeting,
- Discussion points,
- Action list,
- Meeting list,
- Date of next meeting.

4.3.6 Executive summary

Documents of more than 20 pages should contain an executive summary (page 3) following the table of contents (page 2).

5 Project Management Guidelines

The purpose of this section is to provide the Project Management function in general, but especially the project manager, with guidelines to assist in ensuring the highest level of quality in project management.

Project management in general is a science and as such requires clear and precise specification of the activities to be carried out in order to meet the objectives of the project. These guidelines are concerned with the processes necessary for ensuring that the specification of activities is complete at all stages of the project and that those activities are carried out to maximum effectiveness. To that end, therefore, they cover the planning process itself, monitoring the activities of the various work packages against that plan, dealing with any risk to the success of the overall project which may be identified and the controlled resolution of problem situations.

5.1 Project Planning

Many of the problems encountered in meeting the goals of a project may frequently be traced to deficiencies built into the project from its inception. Inadequate preparation often results from a failure of the principles to agree on the objectives of the project, service levels, the scope of the project or sub-components, the level of resources to be committed and any assumptions being made on standards, availability of deliverables, etc.

In the case of CYCLADES, it has to be recognized what type of project is involved. With many partners and many work packages/services, there are in effect a multiplicity of sub-projects all contributing to the whole. It must also be recognized that there are a number of key intermediate deliverables, which are prerequisites for any significant amount of work on certain work packages/services. It is also expected that the boundaries between work packages/services may become blurred as the project develops, but the work must still proceed in a planned and controlled manner.

Each work package/service leader at least should have a detailed plan showing all of the work being carried out and to be carried out by each of the partners involved, showing how they interrelate. It would also seem sensible for each partner to manage his resources so as to make his contribution to the project as cost effective as possible. Similarly the project manager should have a plan which shows the work being carried out by each work package/service to a fair level of detail and how those tasks interrelate.

In order that adequate management of the project can be provided it is necessary that the plans are to a level of detail at which:

- all tasks resulting in the completion of a deliverable whether final or intermediate are identified.
- all inspections or reviews which may be pre-planned are identified.
- tasks expected to last more that 30 days are subdivided where the detail is known.

A project plan identifies all of the tasks required, and it also shows how the project is to develop, relates them one to another, and includes the timing of start and finish.

5.2 Progress Monitoring

It is a truth to say that progress is measured as a reduction in the work still to be done. The expenditure of effort which does not have this effect does not result in progress. Particular marks of progress are the completion of tasks and the shipment of deliverables, for the longer activities it is desirable to identify subtasks so that progress can be measured.

The way in which progress can be measured is thus first of all to define what has to be done. This may well be subject to revision, but such revision should be controlled, agreed and measured in itself. The definition of what has to be done must be documented and be explicit. It is wise to consider that anything which is only implicit does not exist. The documented statement of what has to be done should include a complete definition of the outputs from the project and the companion inputs, the means necessary to enable it to be done, and the means of ensuring that it is done.

A project plan includes tasks to carry out all of these things, and each task should have associated resources and responsibility. Anyone responsible for a task should be required to provide a statement of effort expended on each task on a prediction of the effort still remaining to be done.

It is the project manager's responsibility to see that team leaders (e.g. work package/service leaders) are monitoring progress to a satisfactory level of detail and accuracy. This will require him/her to obtain progress reports on a regular (e.g. monthly) basis and to review the project with the subordinate teams from time to time so as to ascertain whether the reported position is valid and to determine any supportive or remedial action which may be necessary.

5.2.1 Reports

To be of value a progress report must show what progress has been made and what has not. It is not helpful to the project manager if team leaders give an inaccurate or incomplete picture. With careful study of reports it may be possible for the project manager to identify areas for further enquiry before the matter is raised by the team leader. It is even better if the project manager generates a good relationship with the team leaders such that they raise topics of concern themselves, and are encouraged to offer possible solutions also.

Progress reports should be submitted on time and be up to date if they are to be of any value to the person receiving them.

5.2.2 Reviews

A given team may not necessarily be fully cognizant of the concerns of other teams. For that reason the team plan should be reviewed by the project manager with the team leader. This can be an informal review, or a formal checkpoint review. The aim of such reviews is to enable the project manager to make properly informed decisions. Two committees, the Project Management Committee (highest project authority, representing partners interests, major management decisions) and the Project Technical Committee (technical choices, approval of deliverables) will assist the project manager when exploring areas of concern. These committees should be representative of all those involved in the project, and be supportive to the project manager. It does not take away the final responsibility for progress, which rests with the project manager.

5.3 Checkpoint Reviews

The aim of checkpoint reviews is to provide a framework of control for the project manager which assists him in maintaining quality throughout the life of the project. Each of these is aimed at providing a level of assurance that the quality of intermediate and final deliverables is to the level desired. It is not necessary for the project manager to be present at all such reviews, although it is desirable, but it is necessary for the project manager to ensure that the required reviews are held and that any follow up actions are indeed carried out.

5.4 Risk Management

A risk is a potential problem which may or may not occur. It may be identified at any stage of the project and may be related to any aspect. It is however common that they are identified during reviews and in progress reports, because that is the time when the project manager or team leader's attention is drawn to the topic. It is the duty of the project manager to ensure that all such risks are recorded and that the necessary remedial or preventative measures are put into effect.

5.5 Problem Resolution

When problems are identified, they may be satisfactorily resolved if the guidelines below are followed:

The first step is to define the problem. It is well worth defining a problem in writing, so as to determine whether there is mutual understanding of the nature of the problem by those affected.

The second step is to determine whether the situation is a real problem causing immediate impact or is a potential problem which may or may not occur.

The third step is to determine possible courses of action which could remove the problem or limit the damage caused by it. These can then be assessed in terms of effort required, the chances that the problem will be cleared, any wider effects on the project and timing considerations.

The fourth step is to select the actions to be taken and determine who is to carry them out. These should be recorded so that the project manager or work package leader can monitor them to ensure that they are completed.

Finally, if the problem has resulted in a key decision then a decision statement should be raised.

5.6 Decision Statements

A decision statement is raised when a given forum (which may be a work package, a working group, or the PMC) makes a decision which they consider to have significant effect on the project as a whole or their part of it. It provides a means of formally recording an agreement on such a decision.

6 Guidelines for the Inter-relationship between Work Package Groups and Partners

The object of these guidelines is to promote a way of managing the working relationships within and between work packages/services such that there is an improved likelihood of the deliverables meeting the objectives of the project as a whole.

These guidelines are recommendations only, but they are based on accepted best practice in situations where there is a multiplicity of providers and consumers of intermediate products. They recognize the complexity of situations where one producer may have many customers for his products, and many suppliers providing input to the production of these products.

6.1 Definitions

In these guidelines the following terms have the meanings stated:

Customer: A person or group which is the intended user or receiver of an item or service provided by another person or group.

Supplier: A person or group which provides an item or service to another person or group.

Item: A document, piece of hardware, piece of computer software, or other material.

6.2 Principles

One of the principles of quality management is the concept that 'Quality is conformance to the requirements'. This means that all of the requirements, business and technical, of both the supplier and customer are met.

In order for this to be achieved, the requirements which the customer expects the supplier to meet, and those which the supplier expects the customer to meet are defined in writing and agreed between them.

Any individual or group may have several such relationships, and the above principles apply to each and every one of them. Once an agreement has been reached, the supplier has in effect made a promise to his customer. This promise will be expected to be kept.

The second principle, therefore, is that a supplier makes every endeavour to keep that promise, but if he is unable to do so, he should inform the customer without delay and re-negotiate a new commitment.

6.3 Practices

In putting the principles described above into practice, the following process should be followed:

1. The supplier defines who the customers are for each item to be produced.
2. The supplier defines, in writing, his understanding of what the customer(s) wants.
3. The supplier discusses this understanding with the customer(s) so as to make certain that it is correct, or needs revision.
4. The supplier reviews the customer's requirements, as revised, to determine whether he can meet them and reports the results to the customer.
5. Steps 3 and 4 are repeated until there is agreement on what is to be done. This agreement should be recorded in writing.

6. Steps are taken during the period when the supplier is in the process of working to meet the requirements, to ensure that the work will be successful in this regard. These steps include inspections, checkpoint reviews and/or audits.

6.4 Scope of Application

It is emphasized that the key to the success of this process is positive consultation and agreement. The process should be applied to all identified deliverables from one work package to another, and all significant items produced for use within a work package.