REGNET

Cultural Heritage in REGional NETworks

REGNET - Quality Assurance System

Appendix D14

March 2003

Table of Contents

Appendix 1: Naming Convention

This document introduces a convention for naming the documents relating to the REGNET project. This should ease the management of documentation within the project.

Document naming resembles in parts the area of usage of the written piece. The different areas are distinguished:

1. Project Management
2. Technical/Working Documents

All documents of all areas should be named according to the following structure:

RN_CCCCNNNvNN[_AAAA][_<specification>][_f]

Explanation of the naming parts:

Mandatory	
RN	REGNET
CCCC	Codename of the purpose of the document
NNN	Number
v	Version
NN	Version Number (left padded with zero; e.g. 01-09, 10-99)
Optional	
AAAA	Acronym of the Partner sending his contribution to the document; no acronym means all partners
<specification>	Optional text for additional naming for further specification
Final document	
f	Final version

Explanation on version numbering - v - :

The final document (RN_CCCCNNNvNN_f) is only edited by the partner responsible for the document. All the other partners send contributions to the documents - therefore the acronym added.

The versions of partner contributions to documents must always start with v01- their first contribution to the document. Further sending of partner contributions will then have further version numberings.

Naming of Project Management and Meeting Documents

RN_CCCCNNNvNN[_AAAA][_<specification>][_f]

Codenames (CCCC) for the internal reports are as follows:

BR	Bimonthly Report
TR	Trimonthly Report
PR	Progress Report
FR	Final Report
CS	Cost Statement

Exampels:

The first Bimonthly Report sent in by the partner IMAC (Version 1): RN_BR1v01_IMAC

The first official Bimonthly Report integrating all partner reports (Version 1): RN_BR1v01_f

The second Progress Report (Version 3, final): RN_PR2v03_f

The third Cost Statement (Version 1): RN_CS3v01

For meeting management two document types are defined: The Meeting Agenda and the Meeting Minutes. The Agenda will be proposed by the partner organizing the meeting.

Codenames for the meeting documents are:

PMGA	Project Management Group Meeting Agenda (meetings with all partners)
PMGM	Project Management Group Meeting Minutes
PCGA	Project Control Group Meeting Agenda (meetings with co-ordinator/project manager)
PCGM	Project Control Group Meeting Minutes
PTGA	Project Team Group Meeting Agenda (partners working together on task level)
PTGM	Project Team Group Meeting Minutes

Examples:

The final meeting minutes of the first Project Management Group Meeting: RN_PMGM01v01_f

Remarks of partner Zeus regarding the first meeting minutes: RN_PMGM01v01_ZEUS_remarks

Naming of Technical/Working Documents

RN_CCCCNNNvNN[_AAAA][_<specification>][_f]

There are three different types of official technical documents distinguished. The Codenames are:

IR	Interim Report
D	Deliverable
T	Task

Examples:

Interim Report 1.4 (Version 1): RN_IR14v01

Deliverable Number 3 (Version 5): RN_D3v05

The third version of AITs contribution to Task 1.3: RN_T13v03_AIT_contribution

A document by AIT referring to WP1: RN_T1v01_AIT_standards

A contribution to the Deliverable 14 (Version 1): RN_D14v01_docnaming

Appendix 2: Quality Metrics

This section contains some examples and guideline to identify Quality Goals and related metrics. In the following table, there are some basic raw data and metrics:

Entity	Description
Document size	Measured in pages
Code size	Measured in LOC, AELOC
Base code	Code included from a previous release which is not changed in the current release
Delta source size	Added, changed and deleted source size
Total source size	All source code released to the customer; base code, ported code, reused code, new code
Problem	A discrepancy between a deliverable and its documentation, or the product of an earlier phase, or user requirements
Error	A problem found during the formal reviews of the same phase in which it was introduced, but before signoff of the work product
Defect	A problem found later than the formal review of the phase in which it was introduced
Fault	Both errors and defects are considered faults
Mistake	Problem found before a formal review
In process fault	Fault from requirements phase up to but excluding beta test phase of a project. Both product and test faults need to be counted
In process defect	Total number of defects found during development phases.
Post release defect	A defect that is found after release of SW defect

Table 0-1: Examples of Raw Data and Quality Metrics

Two examples of Project Quality Metrics

An example of attribute related to project quality (versus product quality) regards the Effort. Effort towards a project is staff-days a person works on a project. It includes:

- Days spent on training
- Days spent on travel for a project activity
- Days spent doing work for a project

In summary, all working days costs towards the project. Vacation, weekends are the only excluded items.

Another instance is the Cycle Time. Cycle Time is measured as “end date” – “start date”, where start date of a project includes:

1. Date of start of project costing - the day the customer starts to pay

2. Development start - the day the team starts works on the first design requirements, (i.e..) when requirements are baselined

Whereas, end date of a project includes:

1. End of development: release for system integration at customer site
2. Release date: project accepted by the customer, start of post-release
3. Date of retirement: project withdrawn from support (end of maintenance)

Guidelines for counting code

The most common size and normalizing metric collected is the number of lines of delivered source code, because it is relatively well defined and generally accepted.

The Metrics Working Group (MWG) has published an extensive guidebook to counting lines of code. Most of the ideas are common sense:

- Count all active elements of the code that affect the functioning, such as executable statements, data definitions, and labels.
- Do not count comments, program titles, author's name, change history, or documentation elements. (It is possible that separate metrics might be collected for each of these components)
- Reused components, such as macros, library functions, or subroutines, may or may not be counted, depending on the emphasis of the metric.
- A "line of code" implies that each physical line contains a single executable function.
- The contents of "header" or "include" files are counted only once.
- Only "delivered" code is counted. Temporary code written to aid development or testing is not. (It is possible that separate metrics might be collected for each of these components)
- Different programming languages are normalized to "assembly language equivalent" lines of code.

The following are counted as product LOC: data definitions, executable statements, data declarations, subroutine, macro calls (no of times), subroutine, macro code (only once), labels, logical delimiters (begin, end, {, } etc), header and include files (only once), header and include directives (no of times). Note that, in this context, lines are physical lines of code, not logical statements.

The following are not counted as product LOC: temporary code for purposes of development, temporary code for testing, and blank and comment lines.

The following table shows the Assembly Equivalent LOC (KAELOC) conversion for different programming languages. It was determined using the table created by Capers Jones:

Language	Conv. factor
4-GLs	16
Assembler	1
C	2.5
C++	6
ESQL	25
Pascal	3.5
Java	6

Shell	15
Spreadsheets	50

Table 0-2: Examples of Conversion factors

For systems undergoing maintenance rather than development, a “delta” or number of lines of code changed is a more appropriate measure. The delta source size counts:

- Lines of code added to the system.
- Lines of code deleted from the system.
- Lines of code changed.

Software Reliability

Software reliability is defined as the ability of the software to perform a required function under stated conditions for a stated period. It may be estimated based on the number of failures that the software system has experienced since starting testing.

The calculation of metrics related to software reliability is not trivial. The accuracy of results depends on several factors, including the level of detail of the collected data, the way that we select test cases to run, and the particular reliability model used. Software reliability models can be used for making management decisions such as when a product is ready for release.

Important reliability aspects to measure are:

- The change of the software system failure rate over time.
  This is used in combination with a software reliability model to obtain information related to the three aspects listed below:
- Expected additional number of failures to reach a specific failure rate objective (or alternatively, number of defects remaining to be found)
- Expected additional testing time required to reach a specific failure rate objective (or alternatively, testing time required to reach a quality level expressed as the number of defects remaining in the system)
- Expected number of defects likely to be seen by the customer population in a given period of time (for example, over the first year of the product’s use)

The Failure Rate (FR), when associated with the testing time for reaching that Failure Rate, is an indication of the current level of software reliability. FR is defined as the number of failures over the time.

Goal Question Metric (GQM)

In the goal-driven software measurement methodology, business goals are translated into measurement goals by first identifying high-level business goals and then refining them into concrete operational statements with a measurement focus. This refinement process involves probing and expanding each high-level goal by using it to derive quantifiable questions whose answers would assist managing the organization. The questions provide concrete examples that can lead to statements that identify what type of information is needed. In originally devising this measurement scheme, Basili emphasized the importance of having a purpose for the measurement data before selecting data to collect. Without a purpose, we cannot know what the “right” data would be.

GQM examples

Goal: *to improve project planning*

Questions:

1. What is the accuracy of estimating the actual value of a project's schedule?
2. What is the accuracy of estimating the actual value of a project's effort?

Metrics:

- Schedule estimation accuracy
- SEA = Budgeted Cost of Work Performed /Budgeted Cost of Work Scheduled
- Effort estimation accuracy
- EEA = Actual Cost of Work Performed / Budgeted Cost of Work Performed

Goal: *to improve fault containment*

Questions:

1. What is the currently known effectiveness of defect detection process prior to release?
2. What is the currently known containment effectiveness of faults introduced during each constructive phase of SW development of a particular SW product?

Metrics:

- Total Defect containment effectiveness
- TDCE = Post release defects / (Pre + Post release defects)
- Phase containment effectiveness
- PCE = No of errors / (No of errors + No of faults) for that phase

Goal: *to decrease fault density*

Questions:

1. What is the normalized in process faults, and how does it compare with the in process defects?
2. What is the currently known normalized defect content of SW delivered to the customers?
3. What is the currently known normalized customer-found defect content of SW delivered to customers?

Metrics:

- In process faults (IPF) = No of in process faults per KAELOC
- In process defects (IPD) = No of in process defects per KAELOC

Goal: *to improve customer service*

Questions:

1. What is the number of new problems opened during the month?
2. What is the total number of open problems at the end of the month?
3. What is the mean age of open problems at the end of the month?
4. What is the mean age of the problems that were closed during the month?

Metrics:

- Mean Time To Close (MTTC)= Average, for all defects closed during reporting period, of (Date and time closed - Date and time opened)
- Average Age Open (ACP)= Average, for all currently open defects in reporting period, of (Current

date and time - Date and time opened)
Goal: *to improve SW productivity* Questions: 1. What was the productivity of SW development projects (based on their source size)? Metrics: • SW Productivity = Code Size Delivered / Total Development Effort

Appendix 3: Preview Reports / Task Brief

The purpose of this guidelines are to define a framework to apply to the Task Briefs, that according to the Regnet project needs will be considered as preview reports. This guidelines delineate the basic elements to be addressed in the task brief in order to deploy a correct starting point for the activity development.

This should help the project team members (partners) in coordinate their activities, clearly identifying plans, goals, risks, strategy, methodologies… thus assuring from the beginning that the expected results will be achieved.

The preview phase is conducted at the beginning of a phase, for the benefit of the project team, producing a common understanding of the purpose and expected outcome of an activity. All team members shall be aware of the detailed plan of that activity.

The purpose is to focus on the big picture of the current phase, starting from similar projects' experiences, serving as a defect prevention activity. Grey areas and areas of concern shall be highlighted, and risks and dependencies shall be re-assessed, together with the validity of plans. Moreover, the preview is a good place to agree upon standards, conventions, and guidelines.

In order to render previews effective a set of aspects shall be taken into account and have to be reported in the preview reports so that all people affected by them may have an immediate reference whenever needed. In the following the basic elements of these reports will be discussed according to the Regnet project needs.

Key elements

Team Objectives

Purpose: the purpose of the activity to be undertaken shall be clearly addressed. This shall be described in the specific session of the Task Brief Template called "Objectives".

Milestones: according to the duration of the task intermediate milestones may be identified in order to have a coordination point with the rest of the team. Milestones shall take in account as a minimum the Quality Gates and the time needed to rework items identified as needing improvement during the Quality Gate itself.

Deliveries & Output shall be clearly addressed in the Task Brief. In the case of tasks, these have already been defined in the project plan and in the *"Annex 1 - Description of work"*, while for the subtask they shall be clearly identified, pointing out how the sub-task deliverable or output fit into the final deliverables.

Planning

According to planning the following items shall be defined:

Estimation: At the beginning of the project estimates shall be done to identify resources needed to achieve the goals. These estimates shall be reviewed during the project deployment. For the Regnet project precise estimates have been defined at the beginning of the project and according to them resources have already been assigned to tasks and sub-tasks.

Detailed Schedule: as already said for "milestones", a detailed schedule is needed to formalize the different steps the deployment goes through. This is particularly true for those tasks in which different partners are involved at different levels, and in which the success of the activity is strongly dependent on coordination of sub-activities. A detailed schedule shall also take into account checkpoints, time for quality gate and rework. Progresses shall than be tracked against the schedule in order to promptly intervene when deviations from the plan occur.

Critical Path: the critical path may be defined as the set of task that must be completed on schedule if the project as a whole is to be completed on schedule. It shall be accurately identified and tracked to grant that the schedule will be followed.

Configuration Items: items to be put under configuration shall be identified. Examples from the Regnet project are: Task Brief, Interim Report, Deliverables...

Quality Gates: as reported in the Interim Report each task brief shall address the Quality Gate that deliverables or output will undergo (review or inspection), clearly specifying when and who will participate to. From three to five people should be involved in. We suggest that leaders of other sub-tasks belonging to the same task and the task lead take part to the review so that they are aware and agree to the contents of their task deliverable. For code inspection adequate skill and knowledge is required to people who participate in the inspection.

Methods, Tools & Standards

Methods, tools and standards that will be followed during the development have to be clearly identified and described in the "Methodology" session of the task brief.

Risks

Management and Technical Risks: specific risk factors that may affect the success of the task shall be identified at the beginning of the task such as mechanisms for tracking the various risk factors and implementing contingency plans shall be assessed.

Dependencies: linkages between deliverables and tasks shall be clearly identified in order to assure that all affected parties agree with the specific deployment and the overall structure of the part being developed.

Project Goals

Product & Process Goals: specific goals related to the deliverable should be defined since the beginning in order to give an objective way to assess at the end of the task the conformance of what has been developed to initial requirements and specification.

Appendix 4: Task Brief Template

WP Number and Title **Task Number and Title**	**WP x** **T x.y**		
Sub-Task Number and Title	**T x.y.u<.v>**		
Task Leader and Contact **Deliverable Number and Title**	 **D z**		
Interm. Report Number and Title	**IR x.y<.n>**		
Start Date:	**yyyy.mm.dd**	**End Date:**	**yyyy.mm.dd**

Task & team objectives:

- **Objectives:** *describe in this section the purpose of the activity to be undertaken*
- **Milestones:** *according to the duration of the task, intermediate milestones may be identified in order to have a coordination point with the rest of the team (and/or partners). Milestones shall take in account as a minimum the Quality Gates and the time needed to rework items identified as needing improvement during the Quality Gate itself.*
- **Specific Deliveries and output:** *address task as defined in the "Annex 1- Description of work". Clearly identify subtask, if any, pointing out pointing out how the sub-task deliverable or output fit into the final deliverables.*

Description of Work:

- **Detailed Schedule:** *a detailed schedule is needed to formalize the different steps the deployment goes through. This is particularly true for those tasks in which different partners are involved at different levels, and in which the success of the activity is strongly dependent on coordination of sub-activities. A detailed schedule shall also take into account checkpoints, time for quality gate and rework. Progresses shall than be tracked against the schedule in order to promptly intervene when deviations from the plan occur.*
- **Critical Path:** *the critical path may be defined as the set of task that must be completed on schedule if the project as a whole is to be completed on schedule. It shall be accurately identified and tracked to grant that the schedule will be followed.*
- **Configuration Items:** *items to be put under configuration shall be identified. Examples from the Regnet project are: Task Brief, Interim Report, and Deliverables...*
- **Quality Gates:** *task brief shall address the Quality Gate that deliverables or output will undergo (review or inspection), clearly specifying when and who will participate to. From three to five people should be involved in. We suggest that leaders of other sub-tasks belonging to the same task and the task lead take part to the review so that they are aware and agree to the contents of their task deliverable. For code inspection adequate skill and knowledge is required to people who participate in the inspection.*

Description of Component:

Methodology: *Methods, tools and standards that will be followed during the development have to be clearly identified.*

- **Risks:** *specific risk factors that may affect the success of the task shall be identified at the beginning of the task such as mechanisms for tracking the various risk factors and implementing contingency plans shall be assessed.*
- **Dependencies:** *inkages between deliverables and tasks shall be clearly identified in order to assure that all affected parties agree with the specific deployment and the overall structure of the part being developed.*
- **Goals:** *specific goals related to the deliverable should be defined since the beginning in order to give an objective way to assess at the end of the task the conformance of what has been developed to initial requirements and specification .*

Partner	PM	Contact (email)	Obligation
1. AIT			
2. ONB			
3. SR (UniVie)			
4. IMAC			
5. SUL			
6. LMG			
7. NRM			
8. KVA			
9. TARX			
10. MECH			
11. MUS			
12. MOT			
13. SPAC			
14. ALI			
15. CC			
16. IAT			
17. GRAN			
18. ICCS(SUSU)			
19. ZEUS			
20. SI			
21. CERT			
22. VALT			
23. TINC			
Total Effort		Comment	

Appendix 5: Review Process

The purpose of this document is to describe the activities, the roles and perspectives that are the basic elements of the REGNET Review Process.

The steps, roles and perspectives described in this document are relevant for all those artefacts developed within REGNET project that shall undergo peer review.

Instructions

The REGNET Review process is well defined and has three stages:

- Review Preparation;
- Review stage;
- Follow up stage.

Review Preparation

- As the Author decides that the material is ready for review, the Task Leader identifies a Review Responsible, who is in charge of coordinating the Review, assuring that it is correctly performed;
- The Task Leader and Review Responsible select a review team, while the Review Responsible schedules the review deadlines (start date, preparation end date, discussion end date), and decide about the objectives and focus of the review: error finding or assessment of concept/alternatives covered in the document;
- Author distributes the Work Product to be reviewed (Line Numbered If Possible) to the Reviewers with enough advance.
- Review Responsible fills the first part of the REGNET Review Report and contacts the review team members in order to assure that dates are agreed upon, material to be submitted to review is available for all of them, focus and process of the review is clear to everybody.
- The Reviewers study the material and record their comments.

Review stage

- Among Reviewers is identified a Review Recorder who will track all faults and problems addressed during the Review
- Reviewers ask questions about problems found;
- Author answers questions, only to clarify, NOT to provide a solution;
- As faults are agreed upon, Review Responsible takes notes of them on the review reports and classifies them in Major or minor faults;
- The Review disposition is determined:

➢ *Accepted:* inspection identified only minor faults;

➢ *Conditionally Accepted:* Correction of major faults is trivial, or Review Responsible is capable of verifying the fixes;

➢ *New Review:* Impact of the faults is very high and require re-inspection by the whole team (or subset of the team);

➢ *Rewrite:* number of faults is too high and the artifact has to be rewritten.

➢ *Disagreed:* a clear agreement on major issues has not been achieved and review has to be escalated.

- The Review recorder completes the REGNET Review Report.

Follow up stage

- The Review Responsible submits to Task Leader issues that has not been solved during the Review (if any) who will decide on them;
- The Review Responsible fills the REGNET Review Summary, publishes it and related problem report, if needed;
- The Author fixes the errors and defects agreed with the Review Team;
- The Review Responsible ensures follow up, finishes the REGNET Review Report and Summary, signs it, and submits it to the Task Leader, who signs it;
- The REGNET Review Report is finally stored into Project Repository.

Review Roles & Perspectives

All the review process participants must be aware of and able to perform their roles in order to conduct effective reviews. In addition to their technical roles, there is also a need for certain meeting roles.

Meeting Roles

Review Responsible (Moderator):

The Review Responsible should demonstrate both technical and inter-personal skills as well as should show a certain degree of sensitivity to group dynamics. The Review Responsible is in charge of:

1. Organizing the review, setting the agenda, distributing the materials, and scheduling time;
2. Conducting an effective review, ensuring that the procedures are followed correctly according to the guidelines;
3. Keeping the review on track;
4. Making sure that problems and action items are identified;
5. Ensuring that the main focus of the review meeting is on uncovering problems and not fixing them;
6. Making sure that all the relevant forms are completed;
7. Following up to ensure that any problem found is fixed;
8. Reporting the result of the review meeting;
9. Ensuring the completeness and correctness of the problem report if any defect is detected during the review.

Author:

The author is responsible for:

1. Answering technical questions;
2. Fixing faults found in the review and refining the work products according to the review.

Review Recorder:

The Review Recorder cannot be the same person as the Review Responsible and must be able to communicate with the Review Responsible easily. He is responsible for:

1. Being familiar with key words and the notations used in the project so that the notes are easily recorded;

2. Recording problems found in the review and making the notes available to the reviewers;
3. Being able to participate as a reviewer;
4. Polling for consensus, if necessary;
5. Summarizing the results in the REGNET Review Report.

Optional participants:

These participants are responsible for:

1. Evaluating the material, taking care not to overlook any details;
2. Reviewing the p roduct in a constructive manner;
3. Asking questions rather than making accusations;
4. Raising issues, not solving them;
5. Avoiding discussions on style and concentrating on technical correctness and completeness instead;
6. Recording and classifying comments;
7. Sharing in responsibility and ownership.

Fault classification

Major Fault:

One that if not removed from this work product, or in a subsequent work product, could result in a test or field reported problem.

All other faults are Minor:

Comments in code, language, grammar, extra code, spelling, …

General Comments

In order to render Reviews effective they should involve three to five people.

Various technical perspectives must also be represented in each review in order to cover as many as possible of the requirements that the participants in the development process have for the reviewed document. This is done by having different reviewers “represent” the views of the participants who may have an interest in the quality of the reviewed document. The technical perspectives included are those of the customer, system analyst, designer, coder, system tester, maintainer, quality and others.

Appendix 6: Review Summary Report

WP Number and Title	**WP x**	
Task Number and Title	**T x.y**	
Sub-Task Number and Title	**T x.y.u<.v>**	
Task Leader and Contact		
Review Artefact Identification	**D z**	
Review Report Number and Title	**IR x.y<.n>**	
Review Responsible and Contact		

Location	Fault	Classification	Closed by	Verified by
Location line of fault	Short description of fault	Major/Minor	Author	Review Responsible

Total Effort to fix		**Comment**	

Appendix 7: Review Report Template

WP Number and Title **Task Number and Title**	**WP x** **T x.y**		
Sub-Task Number and Title	**T x.y.u<.v>**		
Task Leader and Contact **Review Artefact Identification**	 **D z**		
Review Report Identification	**RR x.y<.n>**		
Review Responsible and Contact			
Start Date:	**yyyy.mm.dd**	**Preparation End Date:**	**yyyy.mm.dd**
Discussion End Date:	**yyyy.mm.dd**	**Follow up End Date:**	**yyyy.mm.dd**
Objectives:			
Disposition:			
Follow up Description:			
Comments:			

Review Team Members

Partner	**PM**	**Contact (email)**	**Obligation**
1.			
2.			
3.			
4.			
5.			
Total Effort		**Comment**	

Review Responsible Signature	
Task Leader Signature	

Appendix 8: Change Request Template

Text in this document, which is colored red, indicates guidance to the auditor in compiling the report, and should be removed before issuing the CR.

1. Request for Modification Information

Change Request ID:	Filled By CCB				
Date Of Request:	DD/MM/YYYY				
Reference		Original		Derived from	CR from which this one originated
Work Area	Work Area of the request for change				
Task	Task of the request for change				
Originator Of The CR	Name and e-mail of the CR submitter				
Organization:	Organization of the CR submitter				
Type of Change:	Problem / Defect / Minor Enhancement / New Functionality / New Configuration Item.				
Principal Configuration Item Affected:	List of the configuration Item involved				
Other CI involved:	List of Other Configuration Items Affected, if known				
Description: Give a detailed description of the Change Request					
Preliminary analysis (if available):					
Severity:	Definition of the CR importance: High/medium/low				
Priority:	Definition of the CR urgency: High/medium/low				

2. CCB Analysis and Resolution

Status:		Approved		Rejected
Work Assigned To:	Responsible for Analysis and implementation			
Description: Give a detailed description of the analysis				
Work Estimate (staff/days):	Estimate of the work size in calendar days for development, debugging, testing, writing the CR, rework, re-testing, documentation, etc.			
Risk analysis:	How far-reaching or "dangerous" are the changes?			
Impact:		Internal		External
Other Products Affected:	Impact (if any) on Other Products			
Work Area Impacted (if any)	Other work area impacted (if any)			
CCB Approval Signatures with date:				

3. Implementation

Start Date:	DD/MM/YYYY
Completion Date:	DD/MM/YYYY
Quality Gate passed:	Yes/No
Note:	

4. Validation and closure

Status:		Approved		Rejected
New Configuration Item Version(s):	List of the new CI Version(s)			
Note:				
CCB Approval Signatures with date:				

List of Figures

Error! No table of figures entries found.

List of Tables