Paper RC03

QualityView - a program database and validation documentation tool

Peter Gerber, Accovion GmbH, Frankfurt, Germany
Michael Ludwig, Accovion GmbH, Frankfurt, Germany

ABSTRACT

During the analysis of a clinical study a large number of SAS programs have to be developed for the generation of derived datasets and production of tables, listings and figures for clinical study reports and summary documents. In this process, program validation is a central point to ensure the quality of programs. Individual validation steps must be documented. In the past, this process was highly paper orientated and time consuming. Validation information had to be retrieved from several sources.
With *QualityView*, a tool developed with Microsoft Visual Basic, we are able to store all validation information electronically in a central place, a Microsoft Access database, and to print the information into standard reports (validation sheets, Microsoft Excel sheets or Microsoft Word documents).

QualityView retrieves all information, which is relevant for the validation, including metadata like location of program, created output files, date of storage, and name of programmer, description of program, etc. from the program header and from a comma delimited file generated by our SAS macro systems. This information is stored, together with program source code and log files, in the database.

Validation of all SAS programs is performed at least by the program developer and documented in *QualityView*. Depending on the required validation model, the program developer can release the program into independent validation. The due date for independent validation can be set and the reviewer is automatically informed. The validation process performed by the independent reviewer gets also documented within QualityView.

The validation process, including correction and revalidation, is repeated until all findings have been resolved. Colors and symbols indicate the validation status of each program. All steps are documented in the database including date and name.

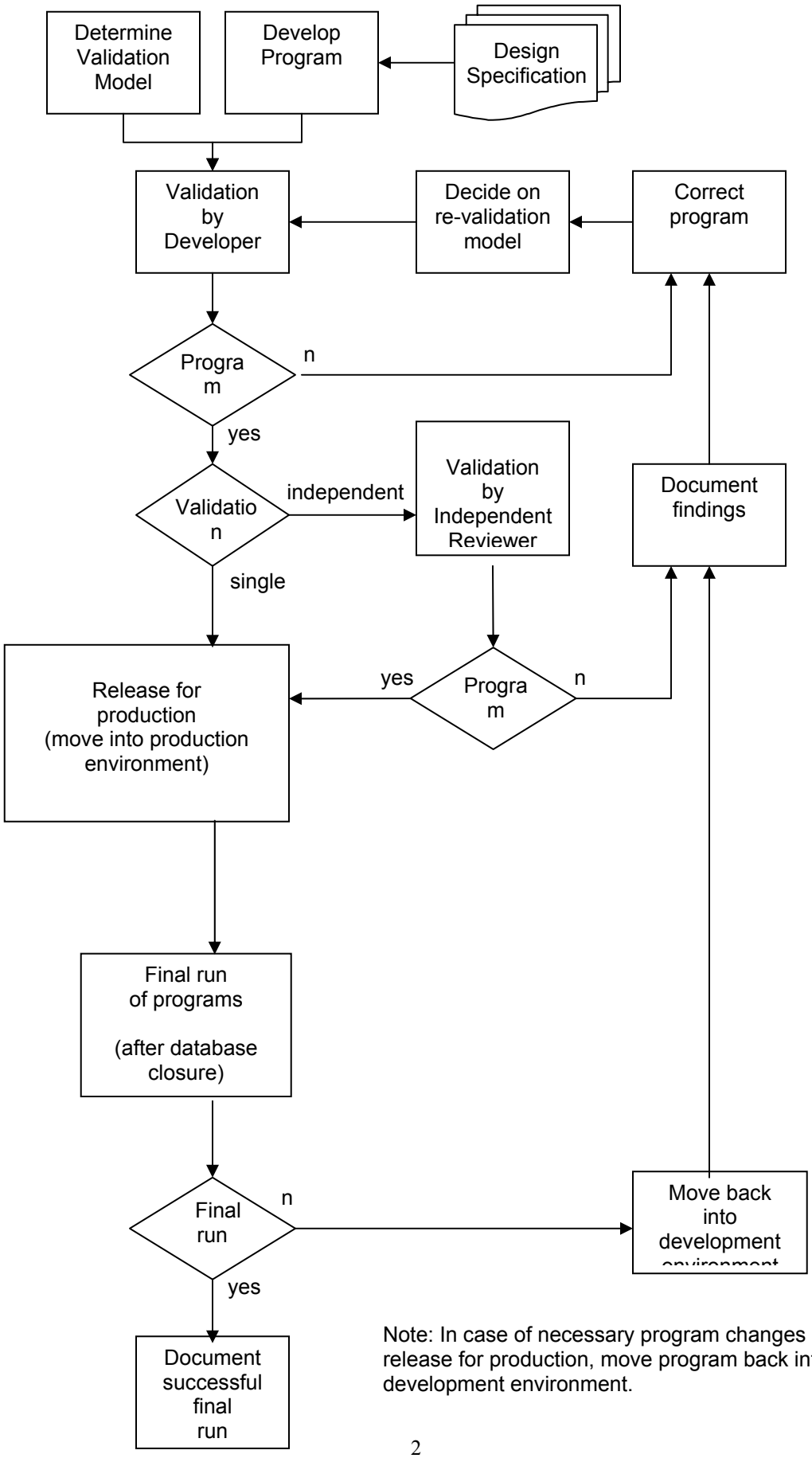
Key words: SAS program validation, independent validation, validation documentation database, validation tracking, validation workflow management

INTRODUCTION

An essential part of any software development is the generation of specifications (e.g. good programming practice guideline, statistical analysis plan, table shells), which need to be defined prior to programming. These specifications along with other guidelines such as standard operating procedures (SOPs) and validation checklists serve as the basis for the validation process.
According to the Accovion SOP for validation of programs, a single validation performed by the program developer is mandatory. The appropriate method for an independent validation can be selected from a range of possibilities (including source code and output review, double programming and verification of test cases) according to the importance of the specific program and the risk associated with a potential program error. Due to the large number of programs that usually need to be validated independently simplification of the validation process and especially of the communication about responsibilities and status within larger programming teams is highly appreciated.

Details of the validation procedure (validation model, responsible persons, test data) together with the results of the validation need to be documented and archived.
Programs are developed and validated in a development environment. As soon as validation is completed, the programs are moved into the production environment.

Figure 1: Accovion GmbH validation flow-chart:



Note: In case of necessary program changes after release for production, move program back into development environment.

REQUIREMENTS AND FUNCTIONALITY

Validation of programs consists of several steps that should all be documented in an appropriate manner. With *QualityView,* a programmer is able to document all these validation steps using only one tool and to store all validation information electronically at a central place. The requirements of the validation process imply the required functionality of *QualityView*.

Additionally *QualityView* has to support our internal processes. For example we have developed a SAS macro system to track the programs and output files (datasets, tables, listings and graphs) that are created for a project and to store this information in a SAS metadataset. This dataset is transformed into a comma delimited file that serves as input for *QualityView*. All programs are provided with a standard program header including a short program description and information about creation and revisions, which can also be used for gathering information.

It was common to collect and maintain the information about program developer, independent reviewer, and further validation information within a list which needed to be created and maintained manually. However, all this information is fed into QualityView and the required documents can be printed at well defined time points and documents are signed by all involved persons.

The main functions of *QualityView* are:

- Retrieval and storage of metadata information. A comma delimited file is created from a SAS dataset that includes information about programs and related output files. Triggered by the programs - specified in the comma delimited file – additional information from the standardized program headers is read and stored by *QualityView*.
  Presentation of the gathered information in an appropriate manner and checking of log files for ERROR, WARNING and NOTE statements, to assist in the validation process.
- Generation of task lists, tracking of validation steps and documentation of primary validation (program developer)
- Generation of task lists, tracking of validation steps and documentation of independent validation (validator)
- Handling of any necessary revalidation steps
- Printing of database information as standard documents (e.g. validation checklists as required and outlined by the Accovion validation SOP which will be printed and signed)

And two additional functions:

- Export an excerpt of the database (e.g. all programs from one study) at the sponsor's request
- Query function (selection of information with SQL) to search and select all programs for special topics across studies (e.g. all programs written by one person, or all programs creating output with demographic data) and to show the selected source codes in an appropriate manner.

DESCRIPTION OF FUNCTIONALITY

Registration of the project:

All SAS programs for a specific study are initally stored under a development environment. The primary validation, independent validation and revalidation are also performed in this environment.
A comma delimited file will be created by SAS to obtain information about the directory path and the created output files for all programs used in the project. *QualityView* uses this information to identify each program's status and to read the information stored in the standard program header.
To update the project information (read comma delimited file and program header) the user clicks the update item of the file menu and selects the interface file in the upcoming file dialog.

```
/*

+ Program name          : ae.sas
+ Version               :
+ Location              : Testxxxx/y001/pg/der
+ Description           : creates dataset dds.ae for Adverse Events
+ Type                  : SAS program
+ Output Type           : dataset
+ Author                : Peter Gerber
+ Date of creation      : 01-Feb-2005
+ Status                : Final
+ Validation            : A+B
+ 1. Validation         : Peter Gerber 22-Feb-2005
+ 2. Validation         : Michael Ludwig
+ Interaction           :
+ Input Files           : dds.meddra_v7_1, dds.demo, dds.medadm, ods.vsdt, ods.ae
+ Output Files          : dds.ae
+ Parameters            :
+ Usage                 :
+ Assumptions           :
+ Limitations           :
+ System                : HP-UX 11.0, SAS 8.2

 Revision History :
  [#]    DATE        Author      Description
  ---    ----        -----       -----------
   1     15-Mar-2005 Michael Ludwig   Change definition of treatment at onset of ae
   2     18-Mar-2005 Michael Ludwig   Check if wardt is missing and reassign correct treatment for onset of ae
   3     21-Mar-2005 Peter Gerber    correct derivation aedurdvc
```

Figure 2: Accovion standard program header

The following summary window lists the properties of the SAS program as read from the program header, including program description. Project and program can be selected from the explorer tree on the left.

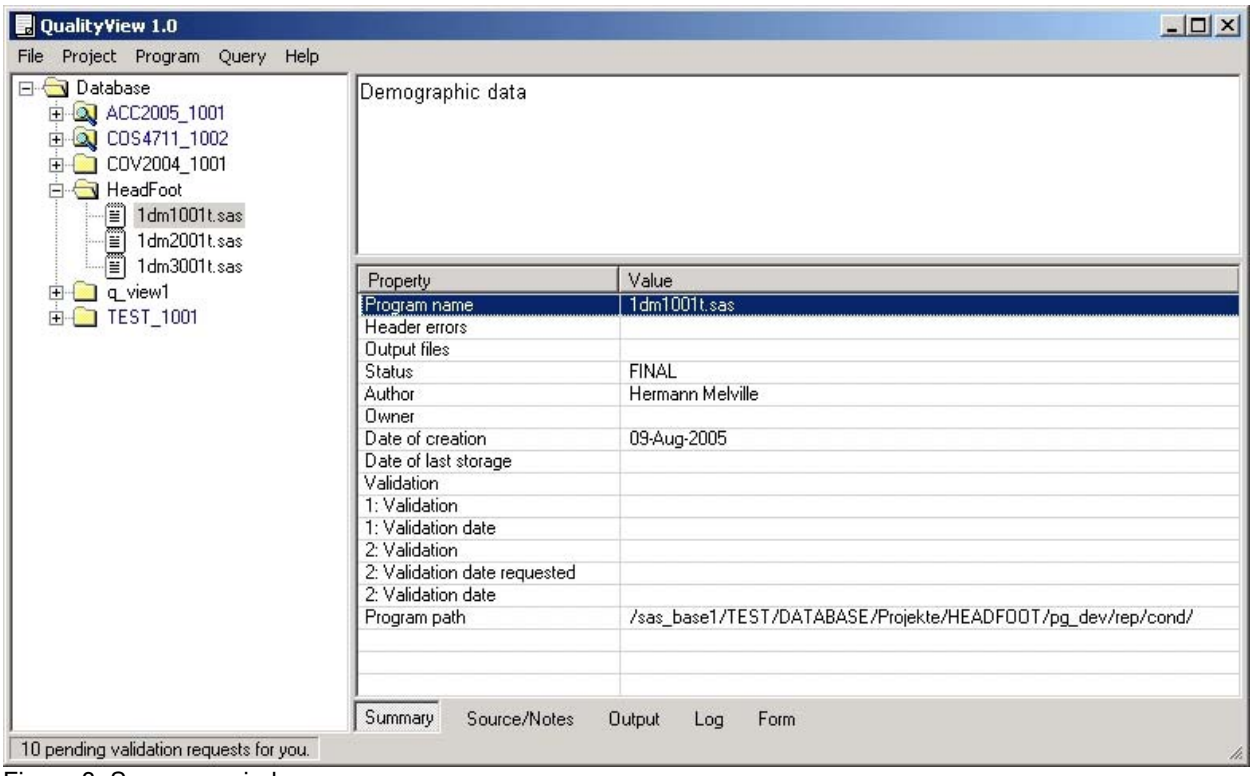


Figure 3: Summary window

With each invoked update, *QualityView* includes the latest SAS source codes in its database (source code/notes window), that thus represents an actual snapshot of the project's respective programming status.

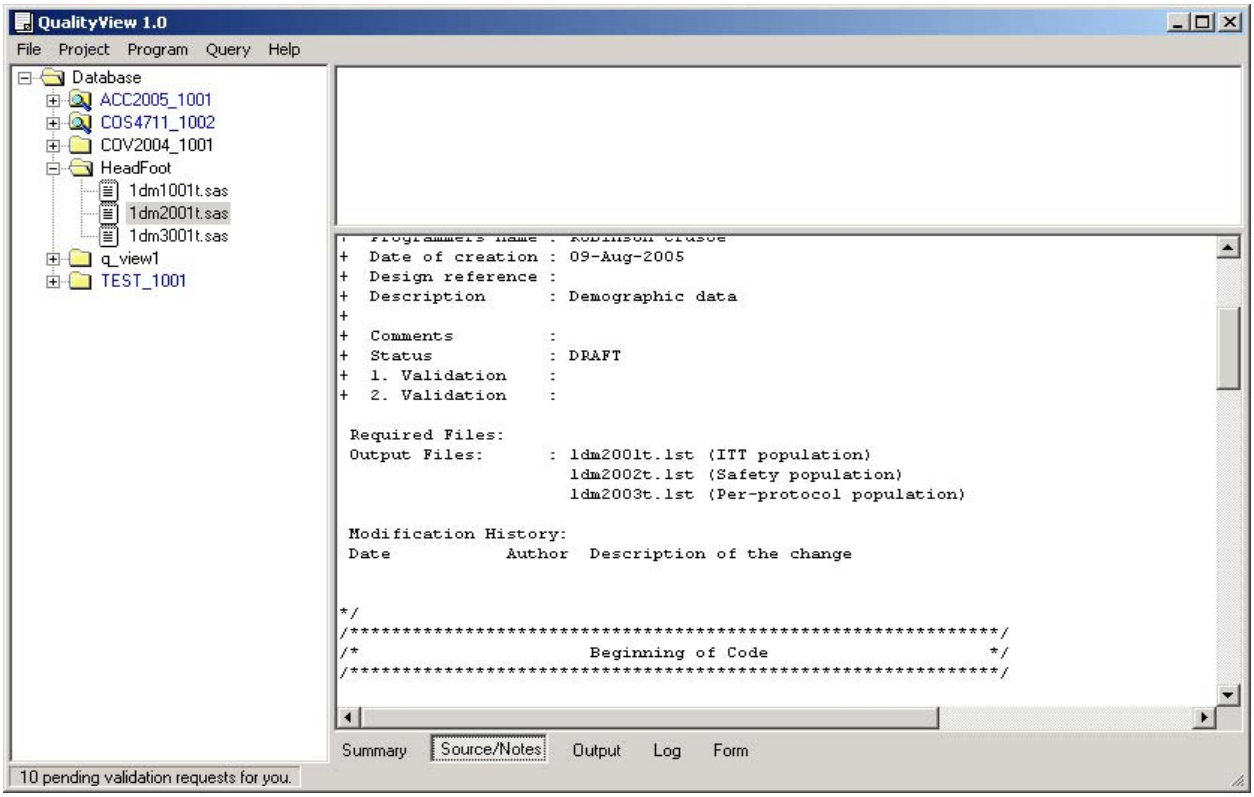


Figure 4: Source code/Notes window

The output files are not stored in the database, but a view to the original file is provided to ensure access to the most up to date version. If a program creates multiple output files, all are presented in the selection tree in the upper part of the window and the user can navigate between them.

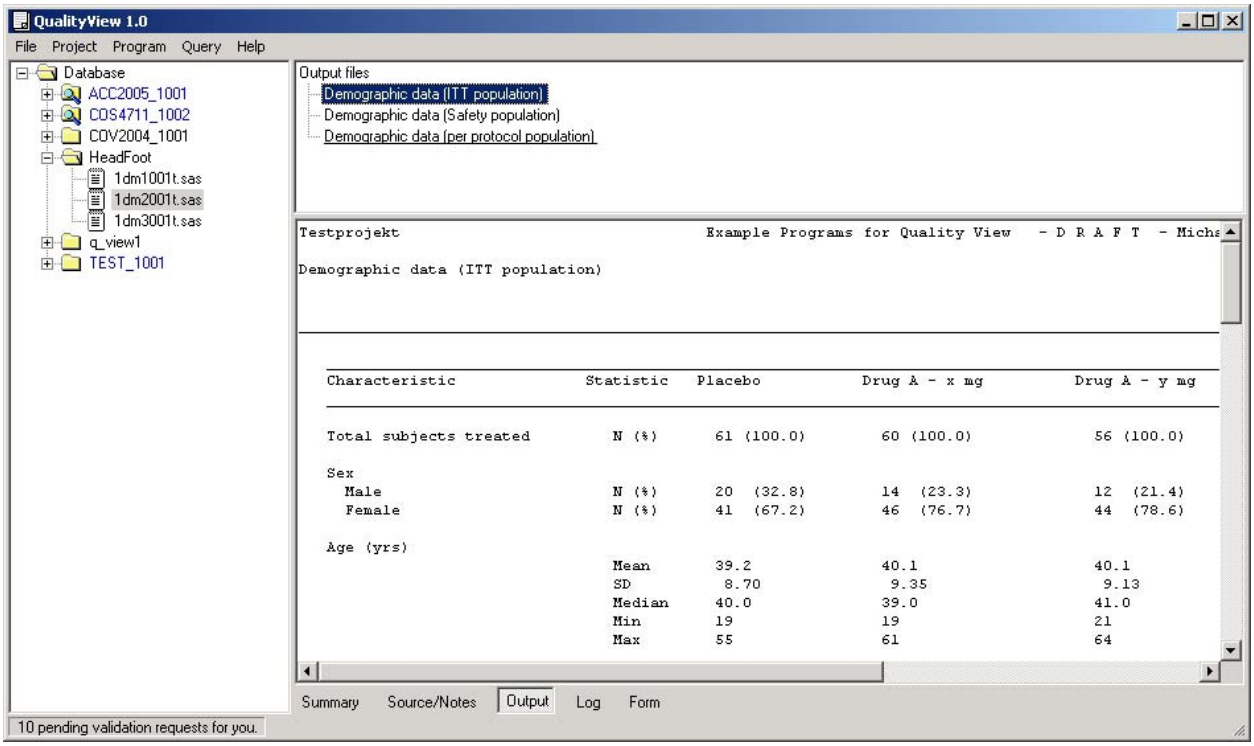


Figure 5: Output window

Additionally *QualityView* stores the log file of the program and searches for key words like warnings, errors etc. (see log window). Findings are highlighted (bold, colored) and kept in the database.

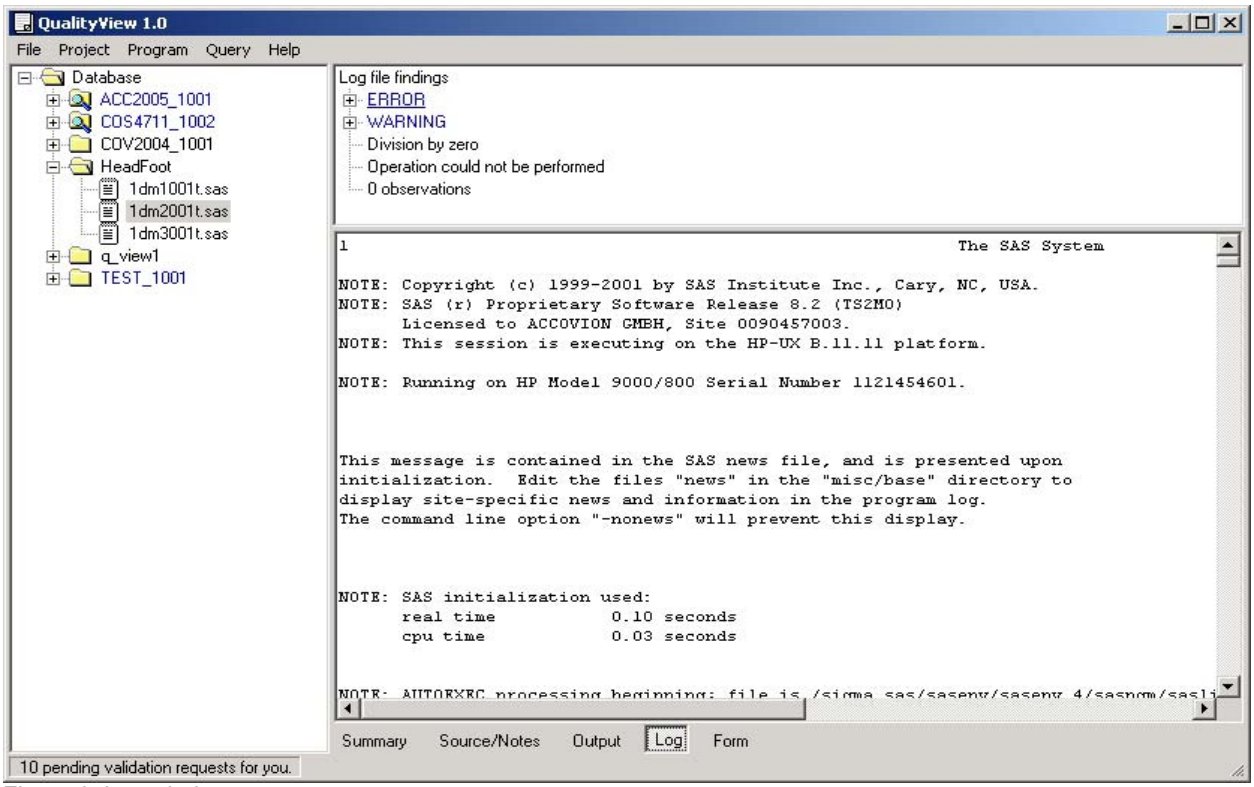


Figure 6: Log window

Primary validation:

When the developer decides that the program is finished, he or she has to start primary validation by checking the program according to the respective specifications and the validation checklist.
The developer starts *QualityView* and selects the program from the explorer tree. With “program/validation” the developer opens the primary validation menu (see next figure).

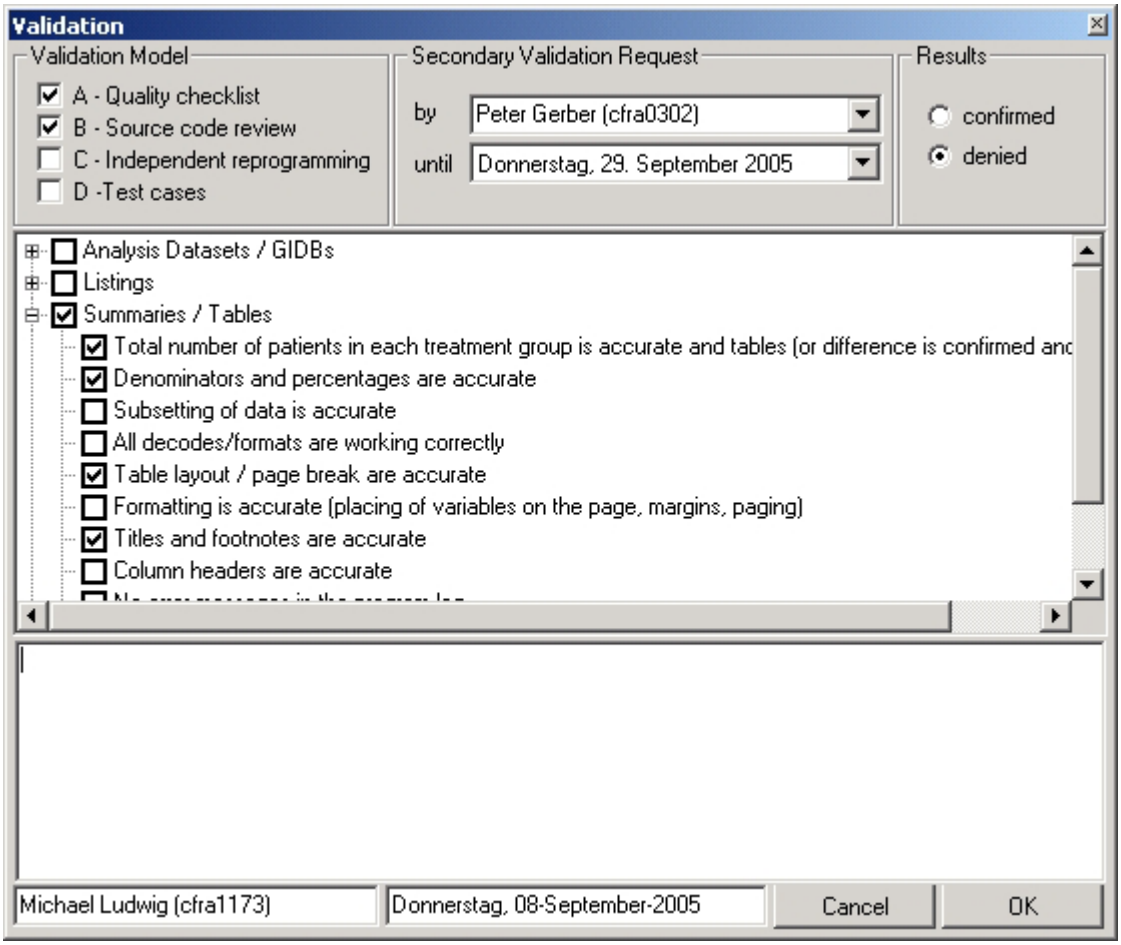


Figure 7: Primary validation window

The developer choses an appropriate validation model, the type of output (tables, listings, figures, datasets) to be checked and, if applicable, selects the independent validator’s name and the desired due date for independent validation and confirms with a click on the validation items that he has reviewed them. In the small window at the bottom of the screen the developer can write a free text comment. If he has not finished the validation, he marks the validation as “denied” and can complete the primary validation later.

As soon as the primary validation is finished, the validation result gets marked as “confirmed”, the program will be displayed accordingly in the explorer tree and the program status changes from DRAFT to REVIEW. The independent validator automatically receives the information upon login, that he has to validate a new program (for example: 10 pending validation requests for you).

Independent validation

The independent validator opens *QualityView* and recognizes the “programs to be validated” through symbols or colors. Then he selects a program and opens the validation window (see next figure) where he can see who has done the primary validation and which items are checked. Then he starts the independent validation and writes comments into the comment window. At the end he has to “confirm” or “deny” the independent validation, which automatically triggers storage of the information in the database. If the independent validation is denied, *QualityView* will treat the program as obliged for correction by the program developer who then needs to initiate a revalidation process. As soon as independent validation is “confirmed” the program status changes to FINAL and the program will have to be moved to the productive environment.

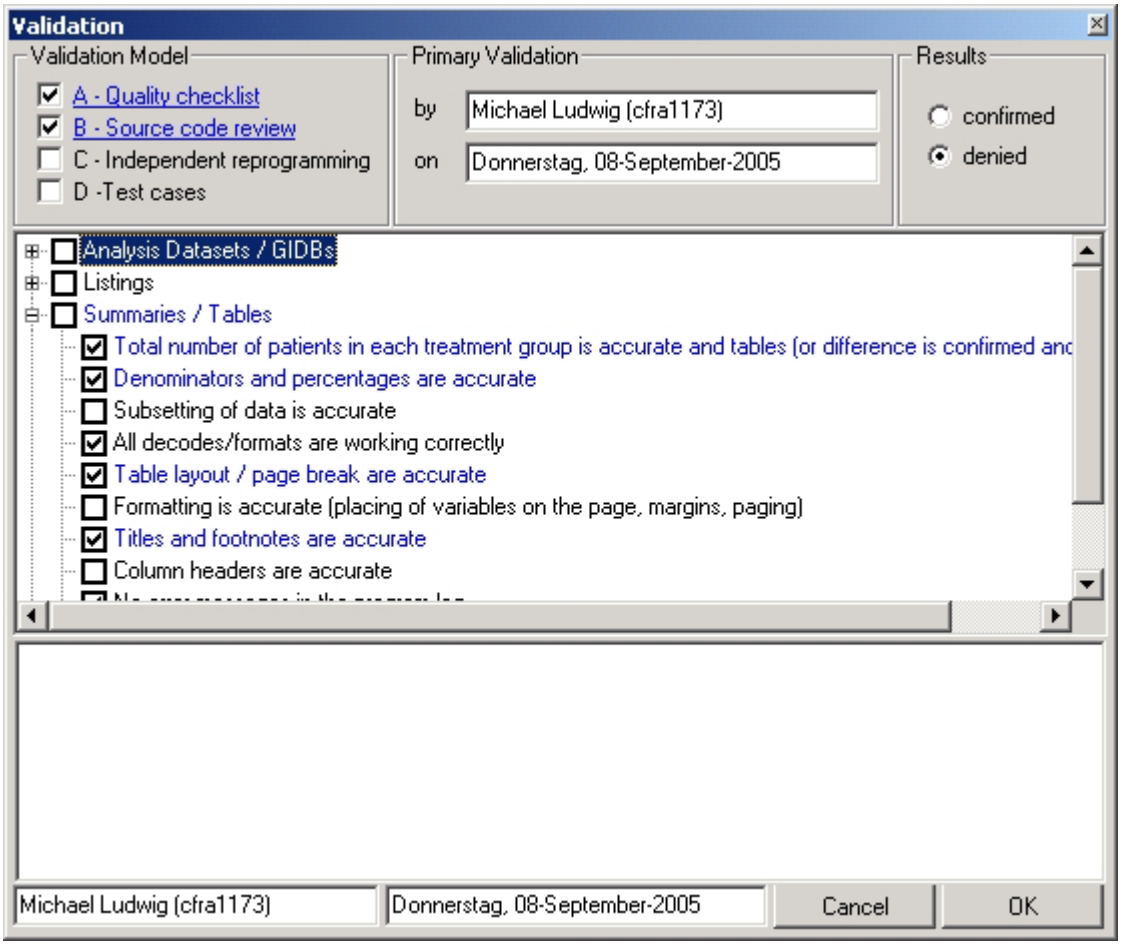


Figure 8: Independent validation window:

The source/notes window shows all of the questions and comments that the reviewer has entered as well as the developer's reply (see next picture). This tracked communication later becomes part of the validation documentation.

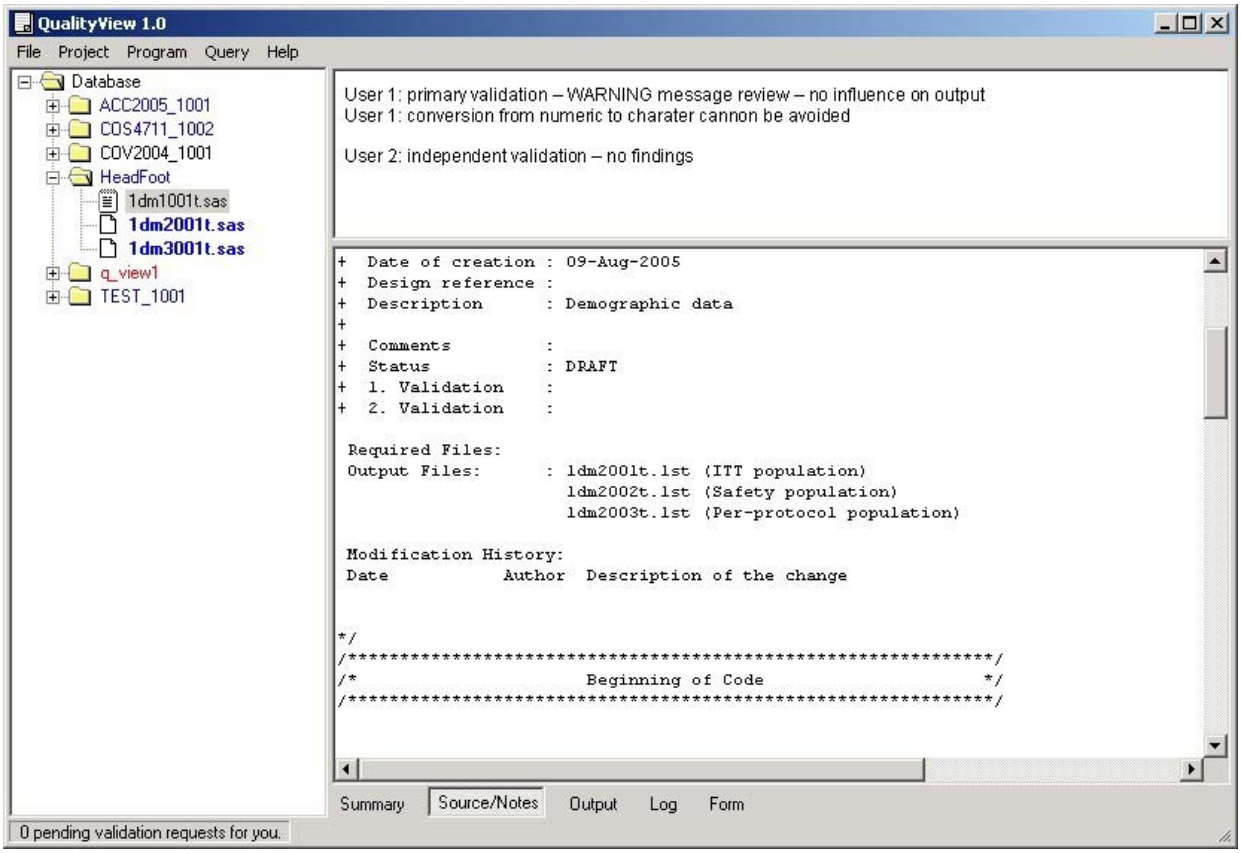


Figure 9: Source/Notes window

Revalidation:

If the validator denies the independent validation, the developer receives information about the findings. He adapts the program and the process of primary validation and independent validation (limited to the required re-validation steps) is repeated. New comments are added to the notes, no information is overwritten.

Printing of validation documentation

All validation information stored in the database can be printed / exported in different validation documents according to the sponsor's guideline. The next figure shows an example of a validation sheet, as created by *QualityView,* which is stored in the database and can be printed easily. The printouts will be signed by all persons involved and serve as validation documentation.

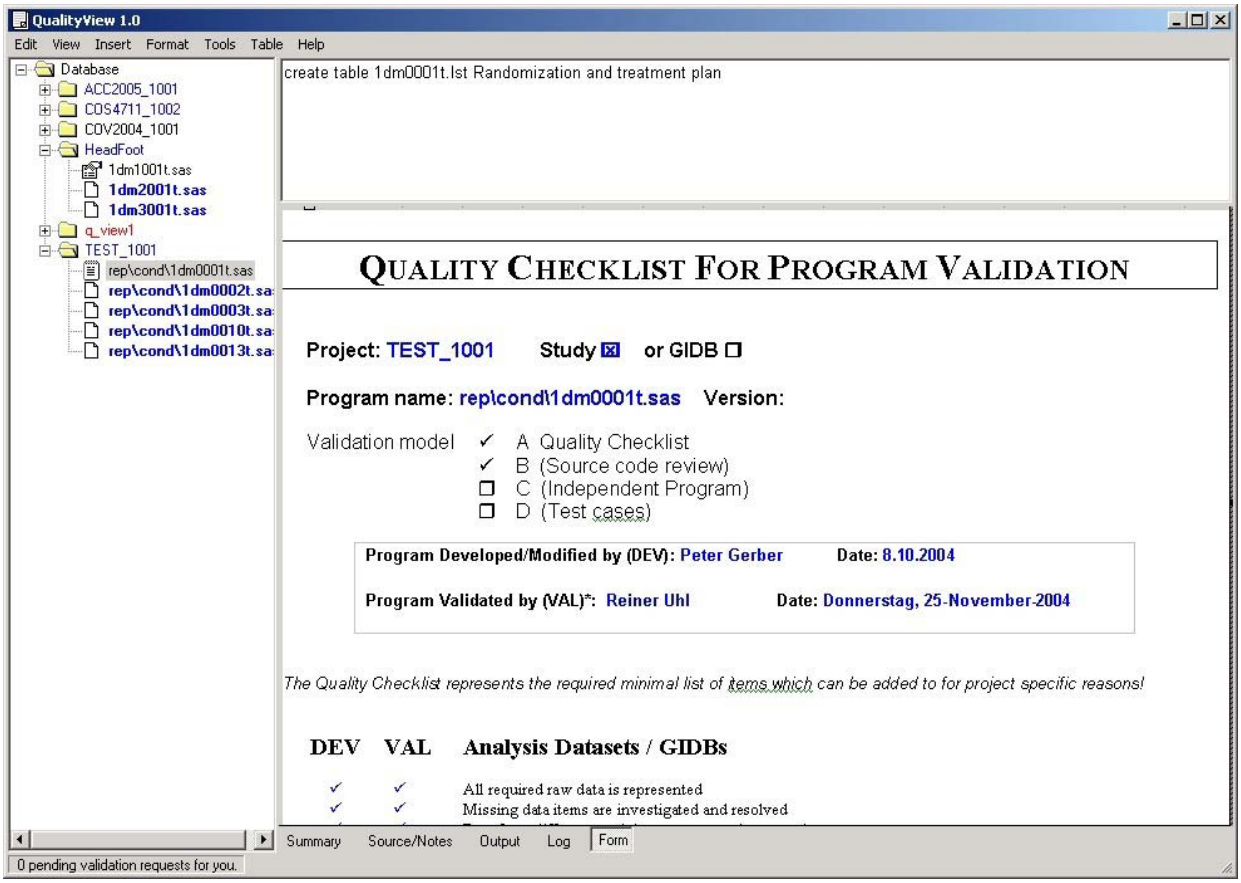


Figure 10: Form windows

Export of database

If requested by a sponsor the database content for an entire project, including source code of all programs, validation information and log-files, can be exported as a Microsoft Access database. A selection of items and projects can be made in the *QualityView* "export client view" window.

Query function:

For the re-use of SAS programs, a query functionality is also included. *QualityView* is able to search programs for a variety of topics, for example all programs written/validated from one person or programs dealing with a special functionality or part of a study (e.g. laboratory data or demographic data).
The source code files can be stored as a text file, printed or adapted to be used in another project.
The implemented SQL functionality provides a variety of possibilities to search and select the validation information, source codes and log files.

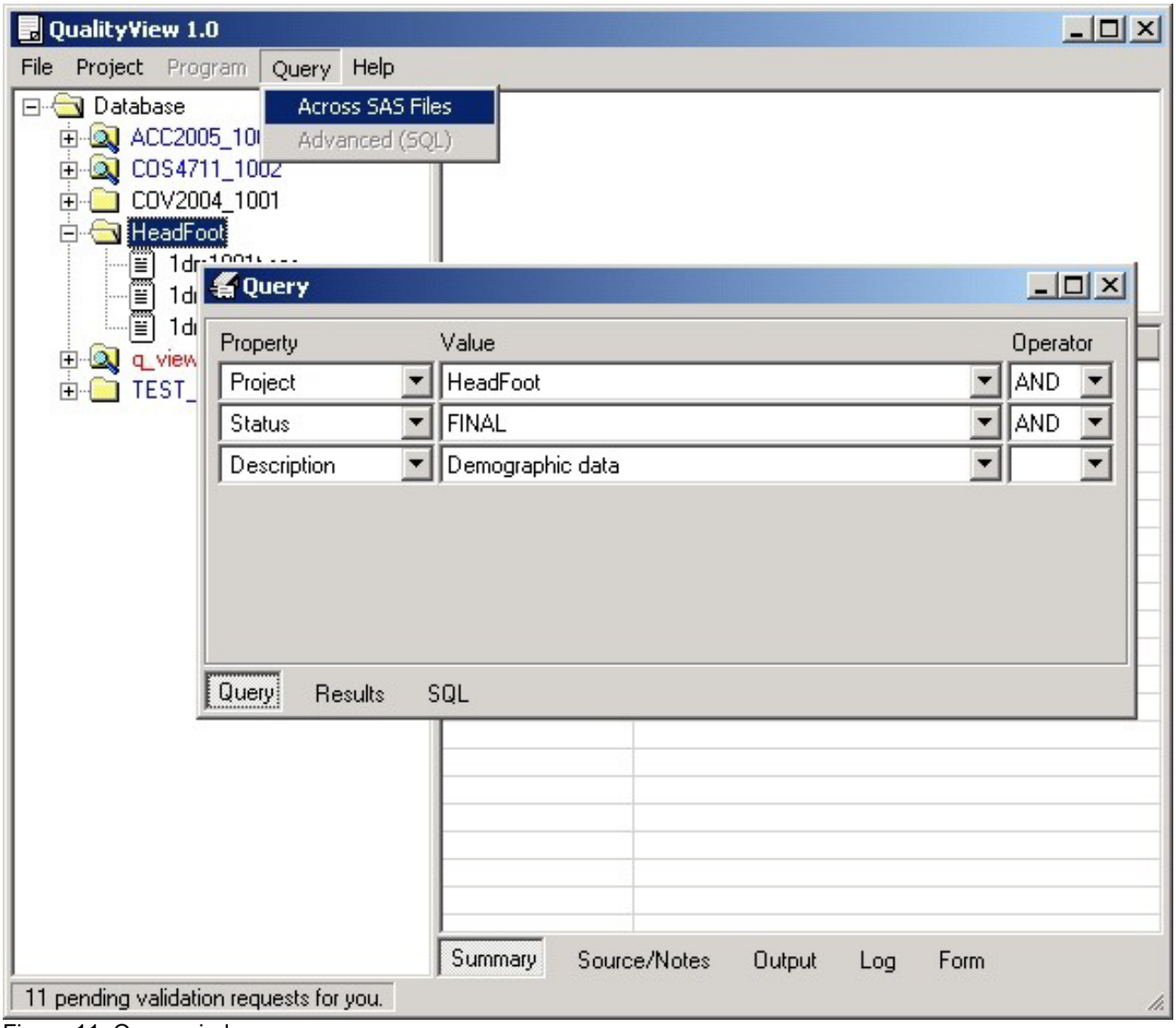


Figure 11: Query window

CONCLUSION

QualityView supports the validation of programs in our standardized process environment by gathering all essential validation information at one central place and offering a menu-driven and self-explanatory platform. *QualityView* keeps all project team members informed upon the current status of SAS programming performed for the study or project and manages the validation workflow by assigning validation tasks to specific team members and alerts them about the most urgent tasks. It always provides an easily accessible and complete overview of all current project deliveries at all stages. It finally presents a complete documentation of all programming and validation steps performed.
Paper documentation of the validation process like validation tracking lists including program names, output files, validation date, name of developer and validator can easily be created from the archived validation history, printed and signed by the responsible persons.

REFERENCES

ACC-SOP-Bio-03 – Validation of SAS-programs for data preparation, analysis and reporting of clinical data Accovion GmbH, Eschborn, 2003
SASEE – SAS software engineering environment, Uhl, Batz, Lembke-Noè, SEUGI, Madrid, 1997

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Michael Ludwig
Accovion GmbH
Helfmann-Park 2
D-65760 Eschborn (Frankfurt)
Germany
+49-6196-7709-319
Email: Michael.Ludwig@accovion.com

Web: www.accovion.com