Declaration of Conformity
Software Validation

Metrohm Ltd.
CH-9101 Herisau
Switzerland
Tel. +41 71 353 85 85
Fax +41 71 353 89 01
www.metrohm.com

825 Lab Link System

Description
The 825 Lab Link System consists of the 825 Lab Link instrument and the PC program 825 Lab Link Server. It allows the Ethernet connection of a Titrando system with Touch Control. Several Titrando systems can transfer their data (methods, results, LIMS data, etc.) to a networked PC with installed 825 Lab Link Server software via such a TCP/IP connection. In this way correspondingly configured folders of the server PC are available to Touch Control as an additional memory medium.
In addition, the Titrando system also has the possibility of sending E-Mail messages via the 825 Lab Link Server.

The **825 Lab Link instrument** was manufactured and tested according to the following standards:

Electromagnetic compatibility: emission
IEC 61326, EN 55022 / CISPR 22

Electromagnetic compatibility: immunity
IEC 61326, IEC 61000-4-2, IEC 61000-4-3, IEC 61000-4-4, IEC 61000-4-5, IEC 61000-4-6, IEC 61000-4-11, IEC 61000-4-14

Safety specifications
IEC 61010-1, UL 3101-1

The instrument has also been certified by the Swiss Electrotechnical Association (SEV), which is member of the International Certification Body (CB/IEC).

The instrument meets the requirements of the CE mark as contained in the EU directives 89/336/EEC and 73/23/EEC and fulfils the following specifications:

EN 61326	Electrical equipment for measurement, control and laboratory use – EMC requirements
EN 61010-1	Safety requirements for electrical equipment for measurement, control and laboratory use

The **825 Lab Link Server software** (version 1) was developed in accordance with the requirements of the ISO 9001 quality system regarding the design, testing and servicing of Metrodata software. The relevant procedures are described in the attached document «Project procedure for creating Metrodata software».

The technical specifications are documented in the instruction manual.
The software was validated with respect to functionality, analytical performance and accuracy of results. The software functions are documented in the instruction manual.

Herisau, 30 October 2002

Dr. J. Frank
Development Manager

Ch. Buchmann
Production and
Quality Assurance Manager

Project procedure for creating Metrodata software

Contents

Project procedure for creating Metrodata software

1 Introduction

1.1 Aim

This document defines the guidelines for the project procedure to be used for creating software. It should result in a controlled and traceable development process. In this way not only the requirements of ISO 9001 can be fulfilled, but the project procedure will have a better structure and the project data will be recorded. This will provide a basis for assessing and tackling new projects in a better way.

1.2 SW development model

1.2.1 Demands placed on a SW development model

The following points must be fulfilled:

- The model should permit a controlled development procedure.
- Guidelines should be drawn up to ensure that certain rules are observed. At the same time these should not be so restrictive that creativity suffers.
- A definition is to be made as to which documents are to be produced at which stage of the project.
- The procedure should be arranged so that as flexible a reaction as possible can be made to additional requests and requirements during the development*.

Various basic models exist, e.g. waterfall, spiral, round-trip form and evolutionary models.

In the QA documentation the use of one of the models per project should be stipulated; this must not explicitly be the evolutionary model. However, as the evolutionary model is the most important development model for us, it is described in detail.

1.2.2 The evolutionary development model (EVO)

EVO is an iterative development process. The targets are defined first and the analysis and design of the complete system is carried out. This is followed by planned implementation steps. The product of each step is made available to the user, whose feedback is included in the next step but one. If sufficient time is available in the procedure then the steps should be planned so that the parts with the highest risk are implemented first. Examples of exceptions are parts whose implementation can be expected to be carried out more easily in future because of the introduction of new possibilities.

* These additional requests and requirements must not concern any primary functionalities or primary requirements!

2 Overview of the SW project procedure

2.1 Activities and resulting products and documents

Either one or more products and documents result from each of the various activities. At the end of each partial project step (milestone) the corresponding products and documents must be available. A milestone can also include an activity, e.g. review of a design.

All important documents such as the requirement specification and the development procedure belong to the version control.

Each employee is personally responsible for the observance of the guidelines.

2.1.1 Obligatory activities/documents

Depending on the size and complexity of a project it may be a good idea to carry out simplifications and omit certain activities and documents.

However, some activities/documents are obligatory. These are marked accordingly in the overview table (section 9).

2.1.2 Project log document

Particular emphasis is given to the project log document. It contains a short list of all activities together with the date and other information (for longer activities the starting and finishing dates). This document must indicate the approximate project status at any time.

3 Requirements

Certain requirements must be fulfilled so that the SW development process can be tackled sensibly. These include:

- A signed requirement specification
- A development procedure containing at least the primary requirements. However, the development procedure itself can be subordinate to the evolutionary model (or any other development model used). This means that independent parts may be at different development stages.

The draft development procedure is to be drawn up by the chemists and then discussed, extended and corrected by the whole project group. The amount of detail contained in the development procedure depends on how much is known about the product under development.

Basically speaking, anything that has not been written down has not been thought out!

Feasibility studies may also be included or possibly a GUI prototype (GUI = Graphical User Interface).

3.1 Defining the project framework

3.1.1 Defining the project responsibilities

The duties and rights of the individual project participants must be defined so that no misunderstandings occur at a later stage. This also clarifies who is responsible for what.

3.1.2 Defining the procedure

Depending on the complexity of the project, the milestones and documents that are to be used in addition to the obligatory ones should also be defined.

3.2 Resulting products and documents

- Requirement specification
- Development procedure
- Description of the project responsibilities
- Possibly a GUI prototype

4 Framework analysis

The framework analysis (analysis of the whole product) consists of two main activities:

- «Matching» the ways of thinking and communicating between the chemists and the software developers
- Examining the functionality

4.1 «Matching» ways of thinking and communication

Misunderstandings frequently occur in projects because the same term means different things to different people. Each project also contains new hardware or software features that have to be given a name and defined.

All important terms should be contained in a glossary (e.g. as an annex to the development procedure).

4.2 Examining the functionality

Based on the development procedure, the functionality of the finished product must emerge from a dialog with the chemists. Here it is also important that the SW developers understand the way that the chemists think and work. It is a very good idea to invest a lot of time in this stage! This is the easiest place to avoid misunderstandings and contradictions in the targets.

Among others, the following aids can be used for examining the functionality:

- Use cases: primarily used to understand the way something works and to envisage typical processes in a better way. If the development procedure itself exists in the form of use cases then this point is not necessary for such a development procedure.
- Object interaction diagrams (business model): simple diagrams that can be understood by everyone. Advantage: gentle transition to object-orientated design.
- Flow diagrams

4.3 Resulting products and documents

- Glossary
- Extended and corrected development procedure
- Analysis papers resulting from the above-mentioned aids

5 Framework design

In the framework design the results of the framework analysis must be converted into a software architecture. If the analysis was already object-orientated then this is more like a refinement of the analysis.

At this stage the division into partial projects (possibly components) must also be carried out. The framework design is not yet concerned with the finest details. However, the interfaces between the individual partial projects must be defined as accurately as possible.

The following points should be kept in mind in the design:

- Flexibility
- Robustness
- Reusability

The degree of flexibility, robustness and reusability is defined explicitly by the definition of the positioning of the product to be created on the market.

5.1 Resulting products and documents

Design papers such as:

- System overview description
- Class, object-interaction, module and Mascot diagrams (there could also be other diagrams, depending on the design method)
- Short written description of the above diagrams
- Description of the project division
- Exact description of the interfaces between the partial projects

A design review is obligatory (to be noted in the project log document)!

6 Planning the implementation steps

The extent of the individual implementation steps and the timetable for them is defined here. The procedure is as described under 1.2.2.

At this time sufficient information is available to estimate times required and, above all, to draw up a timetable. A timetable is essential! This is the only way in which all the project participants can plan how to use their time, and is also the only way in which the project threads can be joined together again at defined points.

6.1 Resulting products and documents

- Planning documents
- Timetable

7 Implementation steps

7.1 Planning

7.1.1 User tasks

The targets of the work must be known before the users can start their tasks.

- What is to be tested?
- Which program parts should be given particular attention?

7.1.2 User feedback

As the user (chemist) is only involved after the first implementation step and the second step runs parallel to the use of the product of the first step, the first user feedback can only be expected before the third implementation step. The evaluation of the user feedback is included in the planning of the next implementation step. It may be necessary to adapt the development procedure as a result of the feedback.

7.1.3 Resulting products and documents

- Target definition for the user task (e.g. test plan)
- Documentation of the user feedback (e.g. bug list and remarks)
- Adapted development procedure
- Design and code adaptations must be planned wherever necessary

7.2 Detail design

The detail design is a fine revision of the framework design for the planned code part.

7.2.1 Resulting products and documents

- Refined framework design
- Short description of the new or extended diagrams

A design review is obligatory (to be noted in project log document)!

7.3 Coding

Coding while observing the Metrohm coding standards (style guide).

7.3.1 Resulting products and documents

- Source files

Possibly a code review for complex program parts (to be noted in project log document!).

7.4 Testing

7.4.1 Module/class test

A test at the module or class level is particularly beneficial for reusable self-contained units.

If several SW developers are involved then at least each part of the implementation step must be tested before integration.

7.4.2 Testing the implementation step

The products of all the SW developers are collected and tested here.

7.4.3 Resulting products and documents

- Test plan
- Test results
- Metrics of the software (if possible)

Possibly a test review as we can learn from each other (to be noted in project log document!).

8 Final testing

The final testing takes place with both the SW team and the chemists.

SW team

As complete a test as possible is carried out according to a test plan with attention being given to the more technical aspects and marginal conditions that would not necessarily be taken into account in a test by the chemists.

Chemists

As complete a test as possible is carried out according to a test plan. The test plan must ensure that all the requirements contained in the development procedure are fulfilled.

The protocol of this test is really our quality certificate!

The final test can, if sufficient comprehensive intermediate tests are carried out, be partially reduced and may only form an addition to the intermediate tests.

8.1 Resulting products and documents

- Test plans (development procedure, etc.)
- Report of the test results

9 Overview tables

Each relevant project step (activity and review) must be listed in the project log document!

The points marked with ★ are obligatory!

Activity	Resulting documents		Document produced by	Review
Requirements	• Signed requirement specification • Development procedure • Description of project responsibilities • Description of procedure • Possibly results of feasibility studies • Possibly GUI prototype	★ ★ ★	• Chemists/Sales/ Development Mgr/Sales Mgr • Chemists/SW Team • Chemists/SW Team • SW Team	No
Framework analysis	• Completed and corrected development procedure • Analysis papers (e.g. use cases) • Glossary		• Chemists/SW Team • Chemists/SW Designer/SW Team • Chemists/SW Designer/SW Team	No
Framework design	• System overview • Class, object-interaction, module and Mascot diagrams (possibly other diagrams, depending on design method) • Short written description of the above diagrams • Description of project work division* • Description of the interfaces between the partial projects*	★ ★	• SW Designer • SW Designer • SW Designer • SW Designer/SW Project Leader • SW Designer	Obligatory
Planning of the implementation steps	• Planning documents • Timetable!	 ★	• SW Project Leader • SW Project Leader	No
Implementation steps	See table below			–
Final testing	• Test plan • Record of test results	★ ★	• Chemists/SW Team • Chemists/SW Team	Recommended
Release	• Release record	★	• Development Mgr	No

* Only if several persons are involved.

Implementation steps				
Activity	**Resulting documents**		**Document produced by**	**Review**
Planning of user tasks	• Definition of objectives or test plan, respectively		• Chemists/SW Team	No
Evaluation of user feedback	• Documentation of user feedback (e.g. bug list, remarks) • Adjusted development procedure* • Description of corresponding design and code adjustments*		• Chemists • Chemists/SW Team • SW Team	No
Detail design	• Refined framework design* • Concise description of the new/revised diagrams		• Programmer/(SW Designer) • Programmer/(SW Designer)	Obligatory
Coding	• Source files	★	• Programmer	Recommended
Testing	• Test plan • Record of test results • Software metrics	★ ★	• Tester/Programmer • Tester • Programmer/SW Project Leader	Recommended

* Only if the user feedback gives rise to modifications