FAST TRAK VALIDATION

UNICORN® Validation Support Package

for

UNICORN® V3.0N

18-1128-30
Edition AA

Amersham Biosciences

TABLE OF CONTENTS

1. INTRODUCTION TO *UNICORN* VALIDATION

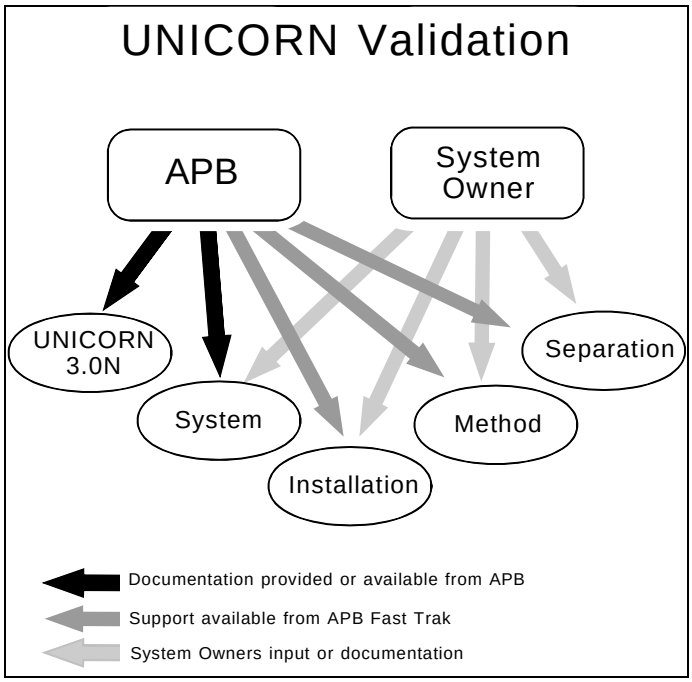
1.1. General

Amersham Biosciences *UNICORN* system affords state-of-the-art control over Amersham Biosciences chromatography systems, providing a level of sophistication and refinement not available in other competitive chromatography control systems. The features and sophistication of this system are such that Amersham Biosciences wishes to protect the system from disclosure to competitors. Despite the superiority of the *UNICORN* system relative to competitive systems, there are always user concerns that control systems meet regulatory agencies (e.g., U.S. FDA) expectations for validated systems.
This support package is valid for all *UNICORN* versions V3.0. The last (third) digit refer to cosmetic changes or bug fixes, and will not affect this document, why it has been replaced with a letter ”N”.
See Section 3, Program Version Policy, page 25.

1.2. Overview

Amersham Biosciences has an unique knowledge how FPLC®, ÄKTA®, BioPilot® and BioProcess® systems have been developed, tested and documented. To system owners, this knowledge and documentation should be of interest in the validation process, assuring that Amersham Biosciences utilizes procedures and documentation which complies with accepted standards throughout the pharmaceutical industry.
The following picture (Figure 1, next page) illustrates the relations between the system owner, Amersham Biosciences and activities and documentation which should be involved in the validation process for a *UNICORN* -based chromatography system.

Figure 1



APB : Amersham Biosciences

Amersham Biosciences refers to the SOPs and procedures, developed and used by Amersham Biosciences in the development of the system.

System Owner is the person at the users site who is responsible for the correct functioning of the system.

Unicorn 3.0N refers to the project specific documentation produced during the development and approval of *UNICORN* V3.0N.

System is the system-specific documentation supplied to the system owner from fabrication and final testing of the individual system.

Installation refers to the Installation Qualification and Operation Qualification (IQ and OQ) for the system. These Guidelines can be ordered separately by the customer, but not as a part of the documentation delivered with the system.

Method refers to the customer's method development and other relevant activities (i.e., education of users) and procedures and their documentation. These activities are described more in detail in Section 6, page 68 (Validation of a *UNICORN*- based Chromatography System) in this documentation.

Separation is the knowledge and activities needed to optimize the process. It involves knowledge needed in the choice of chemicals, media, product materials, buffers. etc. to develop a separation process.

1.3. Intention

This documentation has been assembled to provide important knowledge on the development and documentation of *UNICORN* V3.00. This knowledge assures system owners that *UNICORN* has been developed in a structured manner, and tested and documented in accordance with the System Development Life Cycle concept. The proper development, testing , and documentation are vital to the functionality and performance of *UNICORN* when in operation at a customer's site, and should, accordingly, be of interest to system owners.
It would have been desirable to include all the documents created during the development of *UNICORN* V3.00 in this information package, but this is not possible since the documentation contains information which Amersham Biosciences wishes to keep from disclosure to competitors.

To indicate the existence of this documentation, copies of some pages (as pictures) and partial contents have been included. Section 5, contains the Table of Contents from documents obtained during the development of *UNICORN* V3.00, to indicate the existence of such documents and their contents.

Customers who wish to verify the existence of the documentation and the Standard Operating Procedures used by Amersham Biosciences may perform audits in Uppsala and/or in Umeå, Sweden.

Third party depository of the *UNICORN* V3.00 documentation and source code (also called escrow) can also be arranged when requested by a customer. A standardized escrow agreement for *UNICORN* is available through Amersham Biosciences Fast Trak Validation.

1.4. *UNICORN* development

The *UNICORN* control system was developed within Amersham Biosciences in Uppsala, Sweden, where Amersham Biosciences chromatography expertise was interpreted by knowledgeable programmers to provide a flexible, yet extremely reliable system. Amersham Biosciences utilizes developmental practices and methods to meet the objectives embodied in "Validation Concepts for Computer Systems used in the Manufacture of Drug Products" developed by the Pharmaceutical Manufacturer's Association (PMA) in 1986. Amersham Biosciences staff is current on the issues and expansions of the original concept paper. The concept paper has been utilized as an outline for much of the detailed information that follows. Section 2 of this document outlines the overall validation approach utilized by Amersham Biosciences. This approach is embodied in general SOPs, practices and routines utilized in the design, fabrication, verification and maintenance of computerized systems.

1.5. Functionality

The *UNICORN* system provides superiority of control and flexibility over other systems through the sophistication of its design and configurable nature. Each *UNICORN* system is delivered with the same software, but many of its capabilities can only be accessed if the particular system itself is equipped for that specific function. The sophistication of the software allows even the most complex chromatographic separations to be implemented since they can be assembled from software components included with each *UNICORN*. In contrast to PLC systems in which changes in the purchaser's user requirements require software development, the configurable architecture of *UNICORN* affords a near infinite range of process capability.

The initial development effort including extensive software and system testing performed by Amersham Biosciences on the *UNICORN* system was such that Amersham Biosciences now operates in the operation and maintenance phases of the system development life cycle.Amersham Biosciences does not produce custom software for an individual supplier; standard features in the *UNICORN* software allow custom capabilities without the necessity for further software development. The individual components (instruments) are defined in a "System Strategy" which is a file uniquely defined for each system. The System Strategy specifies the unique templates, definitions of instruments and layout of the individual system for *UNICORN*.
See also Section 4, Configurator.

This document provides a comprehensive description of the validation of the *UNICORN* system that controls *FPLC*, *ÄKTA, BioPilot* and *BioProcess* chromatography systems. It summarizes each of the elements of the validation program for *UNICORN* including system design, software development, hardware qualification and system integration.

The configurable nature of *UNICORN* allows considerable variation in systems to meet the requirements of an individual customer. These activities involve the participation of both Amersham Biosciences and the system's owner from the start of the purchasing activity through system start-up at the owner's location. The steps taken in this area are described in Sections 2 and 3 of this document. Suggested principles and validation order for a *UNICORN* controlled chromatography system are described under Section 6.

1.6. Contents

For convenience in understanding the necessary individual activities, the major elements of the effort are defined below:

1.7. Independent *UNICORN* Audits

When the first version of *UNICORN* was developed, an independent company, Weinberg Associates Inc., audited the principles and procedures employed by Amersham Biosciences in the development of *UNICORN*. As a result of this audit, Amersham Biosciences obtained a certificate which confirms that the development model complies with GMP/GLP and GALP Standards. Copies of the entire document (70 pages) can be supplied to customers on request.

Customers who wishes to verify the existence of the documentation, may perform audits in Uppsala and/or in Umeå, Sweden. Such audits have become more frequent, and many major pharmaceutical companies have sent representatives to verify that Amersham Biosciences meets their expectations for product development activities and quality.

2. CONTROL SYSTEM VALIDATION

2.1. Introduction

The fundamentals of validation for the pharmaceutical industry were first defined in “Validation Concepts for Computer Systems used in the Manufacture of Drug Products”* developed by the U.S. Pharmaceutical Manufacturer's Association in 1986. This landmark article drew upon the pre-existing “software development life cycle” concept and modified it for application within a heavily regulated environment. The “life cycle model" for computer systems validation in the pharmaceutical industry is thus a combination of widely practiced software development concepts and the documentation requirements which permeate GMP compliance activities. The concepts defined in the PMA article have been expanded considerably over the intervening years by numerous contributors.

The most significant refinement of the original effort has been undertaken by the Parenteral Drug Association (PDA). PDA has made two significant contributions to the original effort :

1. Recognition that the correct “life cycle model” embodies revisions to the design details and approach as the development of the system progresses.
2. Awareness that in many situations there are multiple parallel life cycles.

In January 1995, PDA published “Validation of Computer-Related Systems”*. This document is based on the principles mentioned above, but two major changes/additions have been made :

1. It reflects the present technology used in computer controlled systems.
2. It describes the relationship between the owner of a computer related system and the vendor. It emphasizes the importance of selecting a suitable vendor who is ”technically competent and commercially qualified to supply and support the proposed system ”.

This publication is very informative, and can also be used as guidance material for audits. It contains detailed lists (in the appendix) of information to consider for Systems Requirements, Vendor Evaluation and Computerized System Specifications.

Appreciation of the improvements to these original concept papers is important for following how Amersham Biosciences has addressed the validation of its automated systems.

* see Section 7 for references.

2.2. Computerized Systems Validation Requirements

The various activities necessary for the validation of a computerized system in the pharmaceutical industry have been the subject of numerous summary articles. The following text provides a brief description of each of the tasks necessary to complete the validation effort. For comparison to the life cycle diagram in the PMA concept paper, the requirements have been separated into major headings which correspond to the major segments of the PMA life cycle. The following list is not intended to be all inclusive. Depending on the application and sophistication of the system, the details of any individual system may be significantly different from that described herein.

2.2.1. Design & Specification Phase

This portion of the project establishes the groundwork for the entire effort.

2.2.1.1. Functional Requirements

A description of the functions which the computerized system must provide. It defines the scope of the hardware and software required to complete the project. It is essentially a shopping list indicating the desired features of the completed system. It is sufficiently detailed to establish the design while allowing for flexibility in the design at the same time.

2.2.1.2. System Specification

A detailed compilation of the system as designed to meet the elements of the Functional Requirements. It identifies specific hardware features [CPU, printers, CRTs, storage devices, UPSs, and their overall arrangement] and an overview of software considerations [report formats, levels of alarms, scan time, menus, etc.].

2.2.2. Hardware Installation & Qualification Phase

This activity includes documentation and verification of the hardware required. Hardware configuration / specifications for each device should be included. Vendor manuals for each piece of hardware are required. Hardware qualification elements are outlined below.

2.2.2.1. System Description with Schematic Drawings

A schematic representation of the entire system showing the major elements of the system. It should include both process and control system components, as well as any other computer systems with which the system communicates.

2.2.2.2. Piping & Instrument Drawing

Schematic representation of the equipment in the field. There may be as many of these as necessary to depict all of the process equipment. They are generally produced early in the project.

2.2.2.3. Instrument List

A listing of all instruments in the system, including their operating ranges, output signals, selected manufacturer, proposed calibration frequency, accuracy, etc. Each instrument in the list is uniquely identified and can also be found on the P&I drawings.

2.2.2.4. Input/Output List

A list of all signals and inputs received by the system as well as all outputs from the system. Includes communications with other computer systems.

2.2.2.5. Wiring Checks

Verification of the accuracy of cabling and connections to and from all field devices and I/O racks. Also includes verification of cabling between items in the control system and between the control system and other computer systems with which it must communicate.

2.2.2.6. Component Qualification

Documentation that all physical components in the control system are installed in accordance with the vendors recommendations. Includes the satisfactory completion of all self tests possible on each piece of equipment. To the extent that the computer system can be tested without application software, that testing should be performed and documented. Issues such as electrical grounding, isolation from power lines, protection from excessive heat and humidity must be addressed.

2.2.2.7. Calibration

Documented calibration of all field devices from the sensor to the control system. It includes verification that all locations where a variable is stored, displayed or transmitted are within the required tolerances. Includes the completion of SOPs for the routine calibration of the instruments and the initial frequency with which they will be calibrated.

2.2.3. Software Development / Verification Phase

Activities associated with the detailed design, preparation and verification of the software to be utilized on the system.

2.2.3.1. Quality Assurance Methods for Software Development

Copies of procedures utilized by the developers of the program [external and internal] which indicate the software quality assurance measures which are to be followed in the preparation of the code. Such measures

include the use of modular concepts, consistent and logical use of variable names, maintaining documentation on requested changes, etc.

2.2.3.2. Process Operating Logic (pseudo code)

Detailed descriptions of the various procedures to be carried out by the system. It is developed by personnel knowledgeable in the pharmaceutical process for the purpose of communicating process information to the programmer. It indicates the routine process, as well as the desired response to likely process upsets, data entry errors, component failures, etc. Pseudo code can be understood by persons not familiar with computer languages. Logic flow diagrams are sometimes employed for this purpose.

2.2.3.3. Program Description (application Code)

Paper and/or electronic copies of the software which is to be validated. This documents the program and should conform to the pseudo code. It is generally in a computer language. The annotation of source code is highly desirable as a means of initial preparation, audit and subsequent change. At this stage the software can be audited but it cannot be rigorously tested until it is installed on the hardware.

2.2.3.4. Software Structural Description

An overview of the software defining the various programs, including modules [and their functions], their relationship to each other, any sub-routines and sub-programs. It sometimes includes additional information such as protocol lists, batch sequences, alarm points, etc. which are utilized. This information may be assembled immediately after completion of the pseudo code or it may be brought together after completion of the software.

2.2.4. System Integration and Testing Phase

Once the hardware installation and software development is complete, the project enters the integration and testing phase. Here the software and hardware are combined, and testing of the combination is performed. Usually this testing is performed in a modular fashion, but eventually all parts of the system must be placed into service. Validation protocols are prepared for all of the testing performed in this phase. The results of all testing done at this phase are included in formal validation reports.

2.2.4.1. Module Level Testing

Testing of individual modules to demonstrate their adherence to the specifications. Testing at this level consists of individual functions; e.g. alarms, report formats, screen displays, data transmission. Testing of individual modules in a distributed control system is an example of this type of testing.

2.2.4.2. Program Level Testing

Program Level Testing is testing of individual programs to confirm their conformance to specifications. Testing at this level is performed in one or more of the following ways.

2.2.4.2.1. Simulation

Simulation is the use of simulated inputs to the control system to confirm that the **computer system** responds in the desired way to each input. The inputs should be both in accordance with the routine process [uneventful - to confirm proper sequence] or outside the norm [eventful - to confirm the appropriate response to anticipated process or system upsets]. The simulation trial confirms the acceptability of the **computer system**. Simulation should be performed on every module which will be utilized on the system.

2.2.4.2.2. Placebo Batching

Is the use of simulated product to confirm that the **computerized system** performs in the desired manner. The placebo batches should be made both in accordance with the routine process [uneventful - to confirm proper sequence] or outside the norm [eventful - to confirm the appropriate response to anticipated process or system upsets]. The placebo batch confirms the acceptability of the **computerized system**. Placebo batches should be made for every product to be made in the system. In some instances it may be necessary to make both maximum and minimum placebos to assure that process parameters, alarms, and sensor locations are appropriate for all batch sizes. Placebo batching is also used for non-product operations such as cleaning.

2.2.4.3. System Level Testing

Verification of system performance under actual use is called System Level Testing. This is performed at the user's site.

2.2.4.3.1. Product Batching

Batches of actual product are produced in the system using full automation. The batches produced are subjected to full validation testing to ensure their conformance to the required product specifications. No intentional upsets are introduced into the process. The successful production of product establishes the suitability of the system to make releasable material. Comparable testing is performed on other automated activities managed by the control system to confirm their acceptability, i.e., cleaning, data reduction, etc.

2.2.4.4. Additional Software Testing Requirements

Additional testing is required to establish that other software features are correctly written and function as desired. Examples of the types of additional software capabilities which must be validated include:

verification of security measures, confirmation of communication capabilities with other computer systems, verification of data archive systems, and protocol management. In general, all the features and capabilities of the computerized system must be validated.

2.2.5. Operational Phase

The procedures and systems needed to ensure the acceptability of the system over time should be inspected at the time of system start-up.

2.2.5.1. Change Control

Change control procedures must be developed whereby changes in the process, process equipment, software and computer hardware may be evaluated, approved and documented. As necessary, additional qualification and/or validation may be needed to evaluate a change. The procedure should allow for both planned and emergency changes to the system. Emergency change methods are needed when situations requiring immediate action to protect personnel, equipment or product are encountered. Change control procedures must include provision for the updating of pertinent documentation on the system including many of the elements of this document. Records of changes to the system must be kept for the same period of time as any other regular production document.

2.2.5.2. System Recovery Plan

Focuses on the data recovery and system restart procedures. For example, database journaling may be used to keep track of all transaction operations that affect the values of database items. A copy of the software, including essential files, should be kept off-line to be used in disaster recovery. The system recovery plan should address all aspects of recovery from loss of a hard-drive, corruption of a file, or loss of power to the system. Procedures for returning the system to full and proper performance must be in place. Provisions for safeguarding product and essential data must be defined. The validation of these procedures is recommended.

2.2.5.3. Operating Manuals

Operating manuals should be available to the users of the system at all necessary locations. These manuals must be written at a level such that the actual operators can use them. Verification of user manual correctness can be a concern for systems with many features.

2.2.5.4. Training of Personnel

All users of the system must be trained on the various functions they will be performing. All training should be documented.

2.2.5.5. Support Personnel

A listing of support personnel and their responsibilities and qualifications should be included as part of the documentation.

2.2.5.6. System Security

The documentation should describe the physical (hardware) security employed to protect the system, as well as software security. Verification of security measures is strongly recommended.

2.2.5.7. Operating Procedures

Standard operating procedures (SOPs) required for manual steps must be completed. These procedures should include operations performed on a routine basis, as well as procedures needed for occasional use, i.e. calibration preventative maintenance.

3. AMERSHAM BIOSCIENCES STANDARDS AND DEVELOPMENT MODEL

3.1. ISO Certification

ISO 9000 is an internationally accepted series of quality management SOPs. Certification requires rigorous auditing by an independent accredited agency which examines the quality documentation system and confirms that the organization actually operates in accordance with stipulated principles and instructions. The series consists of four SOPs, of which ISO 9001 is the most comprehensive, covering product design, development, production, quality inspection and distribution. ISO 9000 also covers the development, supply, and maintenance of software (ISO 9000-3).
Amersham Biosciences achieved ISO 9001 certification in February 1993.

3.2. Introduction

The development of software within Amersham Biosciences is a structured activity, performed in accordance with defined SOPs that delineate both the overall approach and specific requirements. SOPs which address project management activities provide for consistency of approach, documentation, verification and review. Those SOPs which focus on narrower requirements related to coding [programming] and user interface formats ensure that the completed systems will be both usable and supportable. The use of SOPs of both types in the development of systems ensures that completed systems meet the design objectives and are more easily maintained and upgraded. Adherence to the defined SOPs within Amersham Biosciences is verified in the later stages of system development.

NOTE: Throughout this portion of the document, various lists of SOPs are provided in the various categories. These lists are not all-inclusive, but have been edited to identify only the most relevant SOPs in each area. The included SOPs are those in effect at the time of preparation of this document, and may not reflect the current practices employed within Amersham Biosciences.

3.3. General SOPs

The highest level of SOPs within Amersham Biosciences are those which define project execution methodologies. The SOPs controlling these activities are included in a computerized information sharing system called Amersham Biosciences Management System (BMS).
BMS is a client-server based system, which everybody at Amersham Biosciences with access to the Local Area Network (LAN) can install to access the SOPs. The software used for this system is Adobe Acrobat, which gives users the possibility to search for specific information in the system and to view pictures (such as drawings and flow charts).

3.4. Design Reviews

SOP 7E-0201-70 ed.AC describes the Review and Approval-process used by Biosciences for software products. Section “General Aspects“ from this instruction is below partly quoted.

“The Manager of the Software department within R&D (called SM below) has the overall responsibility for the quality of all software products, as well as their conformance with internal and external design and quality SOPs (where applicable). This means that the SM alone decides if a document or a design is approved or not. The project manager (PM) is responsible for initiating and performing Design Reviews in accordance with the development model. The participant list and timing is decided after consulting the manager of the software department within R&D. This means calling all participants, distributing information at least 5 days before the meeting. Minutes must be taken, and shall be stored as a Design Review Protocol.“

The “development model“ (mentioned above) refers to the “Software Quality Assurance Documentation Flow“, which is described in “System Maintenance and Update Policy”, Section 3.

The SOP further states procedures and criteria which must followed when design reviews are performed. This ensures that high quality software is developed.

3.5. System Maintenance and Update Policy

Amersham Biosciences utilizes structured and documented procedures to provide customers with high quality software. The objective is to produce a user friendly software at the edge of technology containing a minimum of bugs. To comply with this, customer expectations, opinions and other relevant information are gathered together with Amersham Biosciences own development ideas and technical modifications. These are considered when *UNICORN* software is modified.

The following text is illustrated in Figure 2, page 23.
When sufficient information and technical modifications are collected to produce a new version of the program, a Requirements Specification (RS) is assembled. This RS contains all desired functional and other changes for the whole system, not only the software. The RS is used to produce a Technical Specification for the software which describes the changes desired in *UNICORN*.

Based on the Requirements Specification and the Technical Specification for the software, a MMI-prototype (MMI = Man Machine Interface) of the new program is constructed, which simulates the functionality of the new program.
The prototype is then demonstrated to key customers, application experts, etc. Their feedback and comments are used to enhance the prototype. This process can be iterated in several steps. If necessary, the RS is also updated. This way of working close to end users ensures that the software is functionally capable of solving users application problems. It ensures that the software is easy to use and understandable.

The System Design Description (SDD) contains a general description of the modules that need to be modified or programmed in order to obtain the functionality of the TS for the software. It also describes module interaction (logically - not in detail), development tools, target environment, and a lot more vital information.

A System Verification Plan (SVP) and a Software Test Plan (STP) are constructed to be used later in order to verify that the constructed program meets the demands from the RS. The purpose of the SVP is to verify that the software can handle all the key applications when executed in a "real" (wet) environment.

The purpose of the STP is to check that all functionality, as described in the RS, is implemented and works according to the TS for the software.

Based on the SDD, an Integration Test Plan (ITP) is constructed. The ITP describes how the modules shall be tested together when integrated.

The modules and their functionality are described in Module Design Descriptions (MDD). Each MDD generates a Module Test Plan (MTP) that specifies how each module shall be tested prior to integration in the program.

After the above activities are complete, the programming starts. Each module generates one (or more) Source Code File (SCF), which contains the actual program code. Design of code follows certain written instructions called "Coding Guidelines". This ensures that the code is possible to maintain, debug and update.
The functionality of each module is tested according to the MTP. If errors are found, they are corrected, and the MTP is executed once again. When the results of the testing meet those specified (in the MTP), the module is integrated in the program and subject to the integration test according to the ITP. The ITP ensures that all modules work together as specified, and that all data flow between the modules is correct.

When integration of the modules has been approved, the program is tested according to the Software Test Plan (STP) and verified according to the System Verification Plan (SVP).

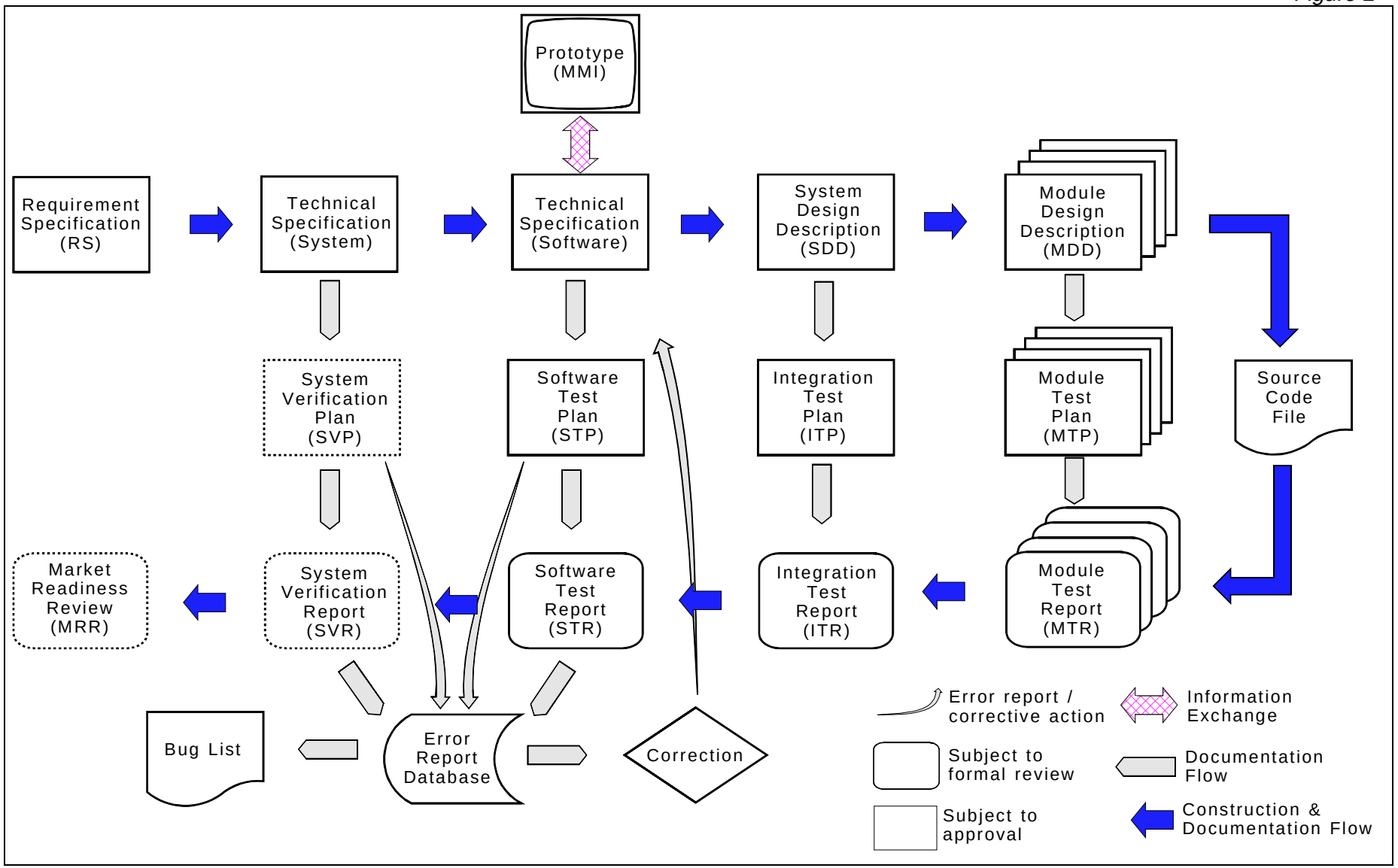
STP testing is done by independent QA-personnel within the software department (not the programmers). SVP testing is done by specially trained application chemists with good knowledge and experience of both the applications and software.

The Market Readiness Review (MRR) is the last link in the chain. During all the MRR all aspects of the new program are considered, These aspects include education, market needs and another aspects relevant for the release of the new program.

The next page graphically illustrates the Software Quality Assurance Documentation Flow utilized by Amersham Biosciences. It is also used in the development of new versions of *UNICORN*.

Software Quality Assurance Documentation Flow

Figure 2



The following SOPs controlling the development of *UNICORN* V3.00 have been used.

Document	Title	SOP Number	Ed.
Requirement Specification	Revision of product Quality	70-7611-19	AA
Technical Specification (System)	Technical Specification	70-7611-09	AC
Technical Specification (Software)	Technical Specification	70-7611-09	AC
System Design Description (SDD)	System Design Description	7E-0202-18	AA
Module Design Description (MDD)	Module Design Description	7E-0202-19	AA
Module Test Plan (MTP)	Module Test Plan and Report	7E-0202-23	AB
Module Test Report (MTR)	Module Test Plan and Report	7E-0202-23	AB
Integration Test Plan (ITP)	Integration Test Plan and Report	7E-0202-24	AB
Integration Test Report (ITR)	Integration Test Plan and Report	7E-0202-24	AB
Software Test Plan (STP)	Software Test Plan and Report	7E-0202-25	AB
Software Test Report (STR)	Software Test Plan and Report	7E-0202-25	AB
System Verification Plan (SVP)	Verification Plan (Software)	BMS doc. spec.*	
System Verification Report (SVR)	Verification Report (System)	BMS doc. spec.*	
Market Readiness Review (MRR)	Market Readiness Review	70-7600-01	AF
Bug List	Bug List	BMS doc spec.*	

* = BMS (Amersham Biosciences **M**anagement **S**ystem) is a database system for document handling, which uses Acrobat Reader.

3.6. Program Version Policy

UNICORN versions are presented as a three-digit number, divided by a point after the first digit.
When *UNICORN* software is changed/modified, the numbering system is used to indicate which degree of modification has been performed.
The following text describes the principles used, and is quoted from SOP 7E-0201-20, ed.AE, which describes the Program Version Policy and principles utilized by Amersham Biosciences.

“The version number used consist of the character "V" (upper case) followed by 3 numerical characters written; **X.YZ**. Example: V1.21.

Increment of X is made when : - major functional changes or additions have been made. - a new hardware platform is introduced. - a new operating system is used. (For changes of operating system *versions*: see below) - other changes related to fundamental safety of the system has been made.

Increment of Y indicates that : - minor functional changes or additions have been made. - file and data structures have been changed significantly. - changes has been made in order to make the software compatible with new (higher) operating system versions.

Increment of Z indicates that : - cosmetic changes have been made. - bugs have been corrected.
When **X** is incremented, **Y** and **Z** are set to zero. When **Y** is incremented, **Z** is set to zero.

This convention must be used for all products that reaches customers.
For version numbering during **product care** activities, the following numbering convention shall be used for "internal" releases :
VX.YZ.Test_Version_NN, where **NN** is "01", "02", ... , "99".
The string ".Test_Version_NN" part of the version number must be removed prior to normal release.
It is allowed to abbreviate the testversion number to VX.YZ.NN, if screen space is limited.
Example : Doing product care on a software with current version number V1.10, aiming at a new release with version number V1.20, will produce internal releases with numbers: V1.20.Test_Version_01, V1.20.Test_Version_02, etc.
When master diskettes are produced, version number is changed to V1.20. This version number must not be used prior to producing master diskettes.
Internal test versions must also be marked with a "message":
"Test Version. For internal Amersham Biosciences use only."
This message must be displayed at program start up, at least 10 seconds.”

OBSERVE !
Upgrading software in a system may have an impact on its functionality, and a revalidation may be necessary. This should be evaluated prior to performing any modification or upgrading of the software.

3.7. System Development SOPs

The development of software and hardware for internally developed systems is performed according to a number of SOPs which broadly define the requirements for each activity.

3.7.1. Software Related SOPs

Amersham Biosciences employs a number of SOPs which relate to programmer activities. These SOPs provide detailed information to programmers regarding aspects of the coding activities. The use of SOPs eases the integration of files into modules, and modules into systems.
The following list describes the SOPs used in the development of *UNICORN* V3.00.

Title	SOP #	Version
Instructions for Storage	7E-0201-13	AC
Service & Production Test Support	7E-0201-15	AB
General Programming Guidelines	7E-0201-16	AD
“C” Coding Guidelines	7E-0201-17	AB
Pascal Coding Guidelines	7E-0201-18	AB
User Interface Guidelines / PC Software	7E-0201-19	AA
Release and Version Control	7E-0201-20	AE
Internal Software Error Report System	7E-0201-23	AC
External Software Error Report System	7E-0201-24	AB
Assembly Language Coding Guidelines	7E-0201-52	AB
“C++” Coding Guidelines	7E-0201-53	AB
Software Design Review	7E-0201-70	AC
User Interface Guidelines / Instruments	7E-0201-72	AB
Visual Basic Coding Guidelines	7E-0202-15	AA
Access Control (uVax/Tek)	7E-0201-10	AB
Backup Routine (uVax/Tek)	7E-0201-12	AB
Access Control (SUN/Microtec)	7E-0202-16	AA
Backup Routine (SUN/Microtec)	7E-0202-17	AA

3.7.2. Hardware Related SOPs

Amersham Biosciences employs a number of SOPs which relate to the design of control hardware utilized in systems. These SOPs provide detailed information to the design staff relative to hardware features, interfaces, etc. The hardware SOPs aid in assuring the operability of software on *UNICORN* based systems.
The following list contains the most relevant hardware related SOPs used in the development of *UNICORN* V3.00.

Title	SOP #	Version
Change Request Order	70-0200-25	AC
Modification Request	70-0200-39	AB
Rules for Article Number Changes	70-0200-74	AB
Design Review (Generic)	70-7611-11	AA
Master Diskette Documentation	70-0201-14	AD
PROM Production	70-0201-44	AB
Master PROM Documentation	70-0201-45	AB
Diskette Production	70-0201-58	AB
PCB Documentation	70-0201-05	AB
Product Marking & Labeling	70-0201-42	AC
Design Review for Electronics	70-7611-61	AA

3.7.3. Production and Testing SOPs

Systems supplied by Amersham Biosciences fall into two major categories: standard systems (those which conform exactly to pre-determined specifications) and non-standard systems (those built to meet a unique requirement [all of these systems are based upon standard systems, and are largely composed of standard components]). The overall methodologies utilized within Amersham Biosciences to deliver both types of systems are defined in appropriate SOPs.

3.7.4. Handling and Access of SOPs

Amersham Biosciences employs computerized distribution and handling of SOPs. This has many advantages over handling and distributing paper copies from an original SOP-binder.

Personnel involved in program development activities gain access to the SOPs by giving their unique identity and a password when logging onto the PC-network. The SOPs are then available from a menu in the wordprocessor program, and can be printed if required.
Since the user's unique identity is registered by the file server, only the persons with authority to update or change SOPs are allowed to modify the SOP files.
The servers hard disk, which holds the SOP files is copied frequently to another media in accordance with a routine to prevent loss of data in case of a hard disk failure.
Outdated SOPs are stored, which gives full traceability, and the ability to determine which SOPs that were valid at a certain time.
This technique (to use electronic storage and distribution of SOPs) has many advantages, such as:

- The distribution is much faster than if paper copies of SOPs are used.
- It eliminates the risk that "old" versions of SOPs are being used, since the version on the file server is the only valid one.
- A high level of security is obtained, since the SOP can only be modified by authorized personnel and ordered only by a person who is System Owner, Project Leader or Department Manager.
- The users can easily search electronically for specific SOPs or SOPs in a subject electronically. It is also possible to search for text strings, not only in a specific SOP, but also generally through all approved SOPs.

The security in the LAN used by Amersham Biosciences is controlled by an SOP in the Handbook used by the IT-department.
This SOP (number 70-7618-20 ed. AC) describes all activities needed, such as Catastrophe Plan, Virus Check, Instructions for LAN Backup, Backup frequency and Administration of User Access.

3.7.5. Service, Maintenance and Spareparts

Service contracts, calibrations, tests, technical support, education, spare parts and other services for the *FPLC*, *ÄKTA, BioPilot* and *BioProcess* systems are available through local Amersham Biosciences subsidiaries. These subsidiaries have a wide experience in providing good service to customers, and use skilled service personnel to provide the best possible technical support for the systems. Most subsidiaries keep their own spare parts in stock, to minimize the time needed to get a malfunctioning system up and running.

To provide customers with the desired services, there are three levels of service contracts available through local Amersham Biosciences subsidiaries.

First level (also called “Gold Service Contract”) will provide “Full insurance” for a functioning system.

Second level (also called “Silver Service Contract“), provides the same services as First level, but excludes spare parts.

Third level, provides one preventive maintenance visit in which the system is serviced and tested.

For more information, please contact your local Amersham Biosciences Representative.

Spare parts are ordered through local Amersham Biosciences branches, with reference to article numbers stated in the documentation provided with the system. As mentioned above, many commonly used parts are stored locally, and can be provided in a short time. Uncommon spareparts which are part of the system due to modifications, or for other reasons not kept in stock locally, are also ordered through local Amersham Biosciences Representatives, which contacts Amersham Biosciences in Uppsala or Umeå.

3.7.6. Customer Interaction

Amersham Biosciences has developed a formalized methodology for interfacing with customers. This provides uniformity of approach and, more importantly, documentation which will ensure that a delivered product will fully meet the customer's expectations.

3.7.6.1. Project Definition and Proposal Phase

This phase embraces all activities carried out prior to the start of actual system fabrication. It incorporates initial customer interaction, scoping of requirements, selection of system options, etc. It requires careful attention to the purchaser's technical requirements to assure that the delivered

system meets their expectations. This is the first and most critical step in the selection of a chromatography system. Amersham Biosciences personnel seek to identify the customer's application, operational, regulatory and economic specifications for the intended system. Application issues include the details of the separation process: chromatography media and column selection, sample and buffer properties, flow rates, etc. Operational criteria include the desired alarms and system responses to them, control system configuration, column packing, cleaning procedures, etc. Concerns relative to standards center on the applicable regulatory or industry standards by which the system will be evaluated, e.g. FDA, EU, DIN, ASME, etc. (Economics relates to the pricing of the system and is irrelevant to this document.)
The conclusion of the project definition and proposal phase is the acceptance of Amersham Biosciences design concepts for the intended system as defined in a written quotation by the customer. If the proposal is accepted the project moves on to the next phase.

3.7.6.2. Detailed Design

After acceptance of the design concepts by the customer, Amersham Biosciences begins the detailed design of the system. At the onset of design, there is a formal transfer of project responsibility from the sales support staff to an internal Project Manager who will oversee the detailed design, construction and testing of the system prior to shipment. Amersham Biosciences formal Quality Control program provides for numerous controls which must be adhered to throughout this entire phase. An individualized Inspection and Test Plan is prepared for each system. This plan defines the tests to be performed and documented in this phase. The design of the system culminates in the development of a Presentation Binder which describes the detailed design of the system to the purchaser. Approval of the detailed design is required before fabrication can commence. Formalized change control involves the customer and begins with this approval.

3.7.6.3. Fabrication

Documents approved for construction form the basis for construction of the system. Internal activities and those performed at sub-suppliers are subject to Amersham Biosciences overall Quality Control program.
All outside manufacture is closely checked by Amersham Biosciences quality personnel and subject to comprehensive documentation which is archived by Amersham Biosciences.

3.7.6.4. Testing and Inspection

Testing of the completed system is governed by the Inspection and Test Plan developed at the start of the detailed design phase. Testing is performed prior to shipment to verify instrumentation functionality and conformance to the customer's requirements. If desired, customers can conduct their own inspection and/or audit of the fabrication and/or testing of their system. Upon satisfactory completion of the system and completion of the required documentation, the unit is shipped to the customer's site.

3.7.6.5. Delivery and Start-up

The culmination of the project is the assistance provided by Amersham Biosciences service engineers at the customer's site to ready the system for initial use. This includes unpacking, inspection for damage, connection to utilities, confirmation of basic functionality of devices and components, initial calibration, etc. These activities are performed in conjunction with customer representatives. Documentation of all calibration and functional testing is maintained by the customer on site. These activities (called commissioning) are described in greater detail in Section 6 of this document.

3.7.6.6. System Validation

Once the customers accepts the system at the completion of delivery and start-up the system owner is responsible for the activities that follow. The system use controls the extent of the validation efforts. This makes it difficult to make recommendations or advise a procedure that can be followed in each individual case. Section 6 in this documentation gives some guidance about how a system may be validated.

3.7.6.7. Operational Phase

Ordinarily, this is the longest phase of the life cycle. During this phase, Amersham Biosciences provides ongoing support to the end user. This support comes in various forms: spare parts, field service support, notification of available upgrades and options, training seminars, etc. Through these activities the user obtains maximum utility of the system over its useful life.

3.8. Problem Reporting

In order to ensure that correct actions are taken when a problem is reported with *UNICORN* software, Amersham Biosciences utilizes an SOP-controlled system to ensure correct handling and feedback to customers at all times.

3.8.1. SOPs utilized

A software related problem may be reported by anyone working with a specific software product. Regardless of where or by whom the problem is discovered, the same error reporting system is to be used. This system is described in SOP no. 7E-0201-24 (External Error Reporting). This SOP is related to four other SOPs :

Software Release and Version Control (7E-0201-20) (see also Section 3, Program Version Policy).
Modification Request Control (7E-0201-21).
Handling of Quality Reports (70-7607-01).
Quality Reporting (70-7607-00).

In the Quality Reporting SOP (70-7607-00), the “User Problem Form” is described. This SOP describes how it shall be filled in where it shall be sent. It also describes the Software Problem Log and how the associated database are managed within Amersham Biosciences.

3.8.2. User Reporting

Users of *UNICORN* shall always report software problems to the local Amersham Biosciences representative for further action. This person is responsible for filling in the “User Problem Form”, and sending it to Amersham Biosciences, Uppsala.

3.8.3. Problem Classification

All reported problems (i.e., software problems) are sorted in the following three problem classes :

A *Major Problem* is one that either causes or directly leads to unavoidable system or application failure, prolonged interruption in system or application operations, or a corruption of data.

A *Minor Problem* is one that does not lead to system or application failure, or a disruption of data integrity. There is, however, some effect on system or applications function or operation.

A *Cosmetic Problem* is merely an annoyance. It may be a misspelled word or an accelerator key not working.

3.8.4. Problem Processing at Amersham Biosciences

Amersham Biosciences processing of problem reports (quoted from SOP 7E-0201-24, External Error Reporting, Sections 5.5 and 5.6) :

In the case of "Major Problems", the R&D Software Department Manager should, after consulting the product responsible within IBO, immediately present a proposal for corrective actions to be taken, to the project board (PMG) head. This must be done within 5 working days. The proposal should also be sent to Technical Marketing in order to prepare for a fast reply to the customer.

The "User Problem Log" is reviewed by the R&D Software Department Manager at least once a month, except July. The R&D Software Department Manager may, based on the total information in the log, suggest immediate action on a specific problem (or problems), and present a proposal to the project board (PMG).

Quarterly, the R&D Software Department Manager analyzes the summary compiled by Marketing and, after consulting the product responsible within IBO, makes a proposal for corrective actions on <u>all</u> reported problems. Decisions are then taken by the project board (PMG).

The R&D Software Department Manager should also, based in the summary, indicate any negative trends (with regard to quality) for any product, to the project board (PMG).

3.8.5. Forward Reporting

The decision taken by the project board regarding a particular problem is one of the following :

No Action. (NA). Problem is unimportant.

Future Version (FV). The problem will be corrected in a specified future version of the software.
See SOP 70-0201-21 Modification Request Control.

Immediate Change (IC). A change procedure has to be started. See SOP 70-0201-21.

Actions decided upon are entered directly into the problem log by the R&D Software Department Manager.

The procedure to assure that the decided actions are performed accurately are defined in SOP 70-0201-21 Modification Request Control.

Problem information feedback to customers is described in SOP Handling of Quality Reports (70-7607-01) and Quality Reporting (70-7607-00).

3.9. Beta Testing of New *UNICORN* Versions

New versions of *UNICORN* are tested by a selected number of key users prior to final release. The intention is to test the functionality and correctness of the new product before the final product released.
The SOP controlling these activities have number 70-7631-52, edition AB, and is titled “Field Tests of Products in Development Phase”. It describes the responsibilities, planning, performing and follow-up of beta-testing of all products.

4. *UNICORN* CONTROL SYSTEM, VERSION 3.00

4.1. Revision History

The following versions of *UNICORN* has been released from Amersham Biosciences (in chronological order) :

***UNICORN* version**	**Released**
V1.00	Dec. 1992
V1.01	Feb. 1993
V1.10	Sept. 1994
V1.11	Dec. 1994
V1.12	Feb. 1995
V2.00	Dec. 1995
V2.01	Feb. 1996
V2.10	Apr. 1996
V2.20	Oct. 1996
V2.30	Feb. 1997
V3.00	May 1998

See also Section 3, Program Version Policy, page 25.

4.2. *UNICORN* Description Overview

UNICORN V3.00 is a complete package for controlling chromatography and other liquid handling systems from an IBM-compatible PC running under the Windows NT operating system. The Software is primarily designed for use with *FPLC*, *ÄKTA*, *BioPilot*, *BioProcess* Systems and with custom engineered separation systems supplied by Amersham Biosciences. For these purposes *UNICORN* is configured for the specific chromatography systems by Amersham Biosciences. This configuration is referred to as a “system strategy”. *UNICORN* may also be adapted on request to work with other liquid handling process units.
Control of chromatography processes from *UNICORN* is based on programmed methods, containing instructions to control the Liquid Handling Modules. Running a *UNICORN* method generates a result file containing monitor data and process documentation. Data can be examined and manipulated using the evaluation module. Additional software features include manual control of the Liquid Handling Modules (LHM), calibration of system monitors and extensive security functions.

The main features of *UNICORN* are :

It provides full manual programmed control of liquid handling systems (e.g., pumps, valves and monitors).
It controls up to four process systems simultaneously, independently or in programmed sequential operation.
It includes calibration capabilities for system monitors (conversion from signal to engineering units for all monitors and transmitter Calibration for certain monitors supplied by Amersham Biosciences).
It provides a dynamic graphical system overview with critical parameter values and full process documentation.
It provides full batch documentation of chromatography processes in accordance with requirements of GLP and GMP (Good Laboratory Practice and Good Manufacturing Practice).
It includes comprehensive data evaluation software.
It is protected by password access control and activity authorization systems.
It allows active processes to be locked for unattended operation without risk of unauthorized interference.
It gives other PCs with *UNICORN* on the same network the same functionality as the local PC.

4.2.1. Network Connected PC with *UNICORN*

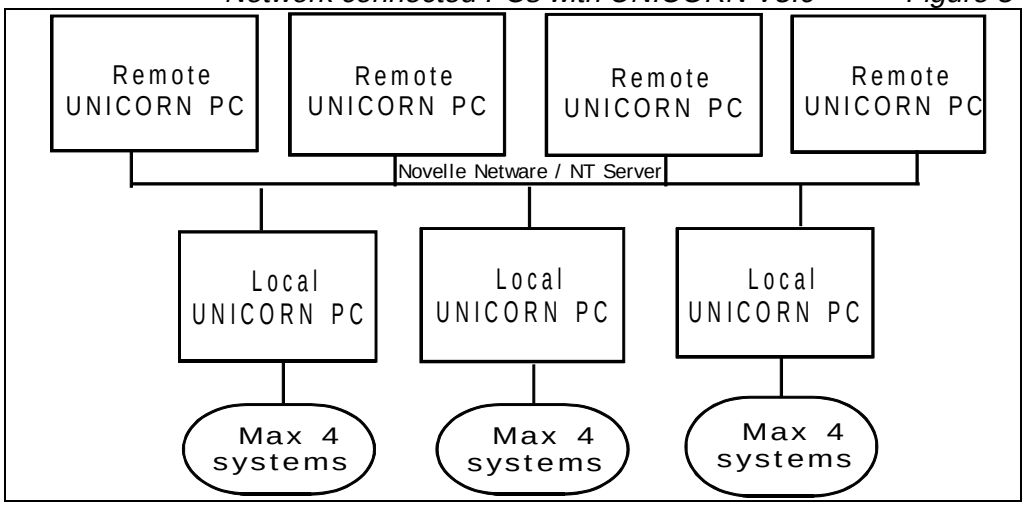
This capability allows remote PCs with *UNICORN* to operate chromatography systems with the same functionality as the local PCs connected directly to the chromatography units. There is no difference in the functionality on a remote computer compared to a local computer.

In a *UNICORN* V3.0N network installation, each of the four System Control modules on each computer (local or remote) can be assigned to any chromatography system in the network. Many System Control modules can be assigned to the same system at the same time.

Assigned system control modules can be reassigned to other chromatography systems in the network at any time, even during a system run.
When a system is assigned, the status, curves, logbook entries and all other data valid for that system are retrieved and displayed as if System Control had been assigned since the run started.

The following picture (figure 3, page 36), illustrates how remote and local PCs on the same network can be connected to obtain the network functionality in *UNICORN* V3.0N.

Network connected PCs with UNICORN V3.0 *Figure 3*



4.2.2. Local PC with *UNICORN*

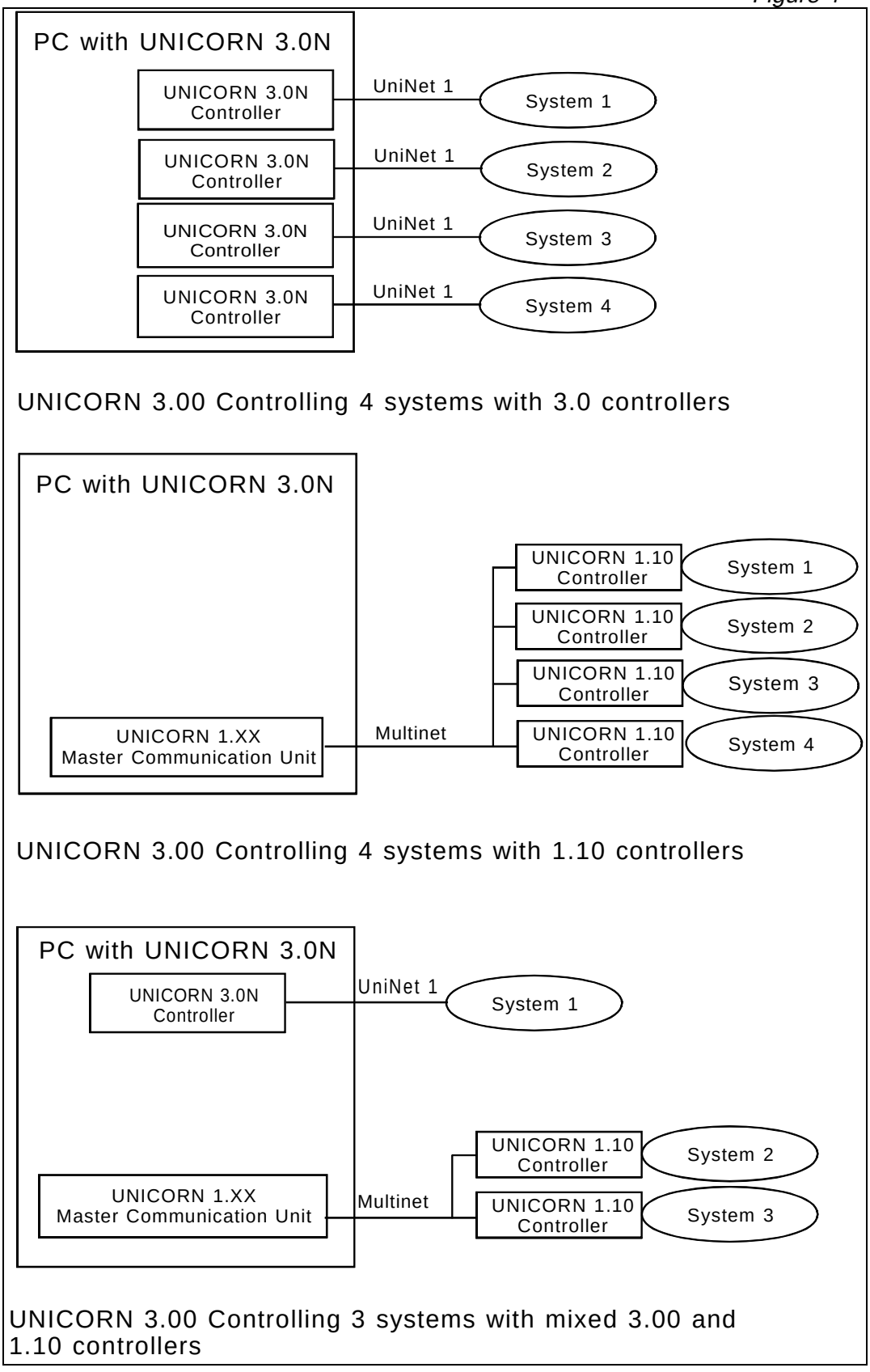
UNICORN can run locally, i.e., the chromatography systems are physically connected to the local PC, or remote. This means that *UNICORN* runs the on a computer(s) in a PC network and the user accesses a chromatography system through a local PC. The local PC may have the *UNICORN* User Interface modules running, in order for the remote *UNICORN* to access the system. The only requirement is that the Open Connection Interface (OCI) is running. From one local PC, *UNICORN* 3.00 can control up to four systems that have:

- *UNICORN* 3.00 controllers on the PC bus
- or *UNICORN* 1.10 controllers in the electrical cabinet of the system
- or a mixture of these

The next figure illustrates three examples of these possible configurations, (figure 4, page 37).

Possible local PC configurations

Figure 4



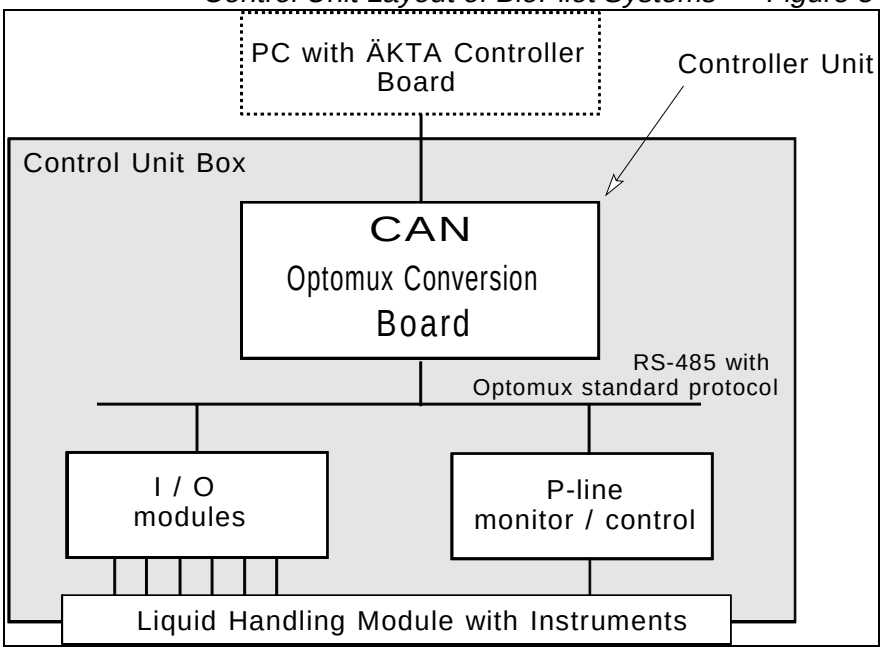
4.3. Functional Description

The whole method is sent from the PC using the CU to the respective Control Unit, which in turn handles the timed execution of instructions and receives data from monitors in the system. The Control Units have their own microcomputer resources that are responsible for controlling the process. Using this functionality, the PC does not have to control the Liquid Handling Modules in real time, and can be used for method development, process monitoring or data evaluation whilst *UNICORN*-controlled processes are running on Liquid Handling Modules.
Communication between *UNICORN* and the Liquid Handling Modules is transparent to the user and can, therefore, be controlled directly with *UNICORN*.

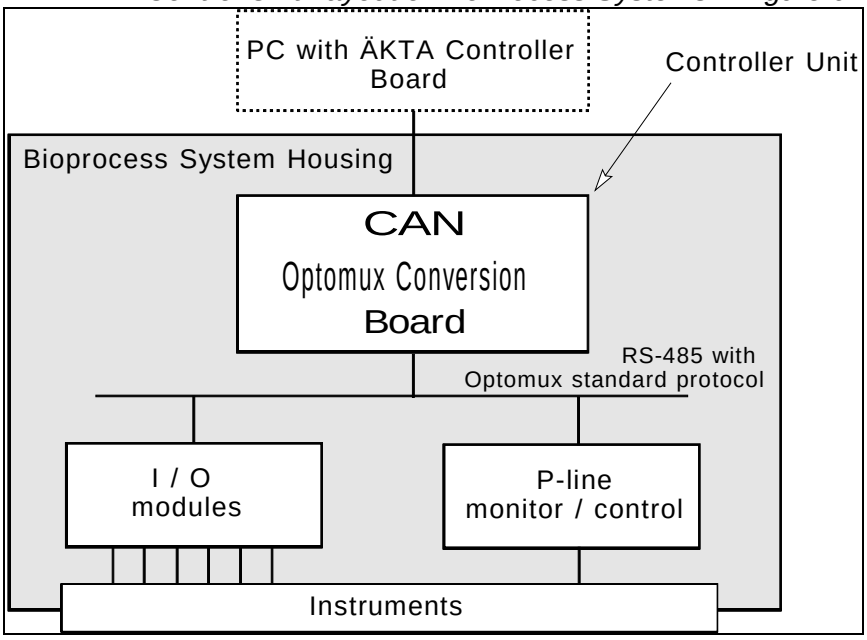
4.3.1. Control Unit Description

Each Control Unit consists of a Controller Unit, I/O-modules and possibly P-line monitor /control boards.
The Controller Unit has a Controller board, Optomux Interface board, and Reset board.
Optomux I/O-modules handle the signal conversion, which is either analog to digital or digital to analog.
The P-line monitor/control boards are instruments developed by Amersham Biosciences (e.g., UV-detector, conductivity meter and pH meter). They communicate directly with the Controller Board in the Control Unit.
In *BioPilot* Systems, the Control Unit is a physical box containing the components described below (Figure 5), while *BioProcess* Systems have the Control Unit components integrated in the system housing (Figure 6).

Control Unit Layout of BioPilot Systems *Figure 5*



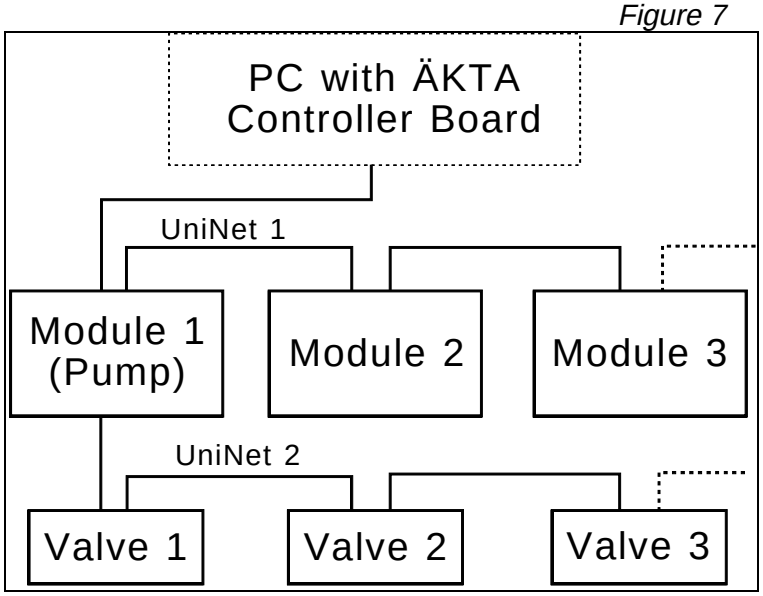
Control Unit Layout of BioProcess Systems Figure 6



4.3.2. *ÄKTA* Systems - Network Description

ÄKTA Instrument modules are controlled from the local PC (or from a PC on the network) by *ÄKTA* Controller Board which is connected to UniNet. The Pump Module has a driver for a second UniNet which is used to control the valves in the system. This is illustrated in Figure 7 below.

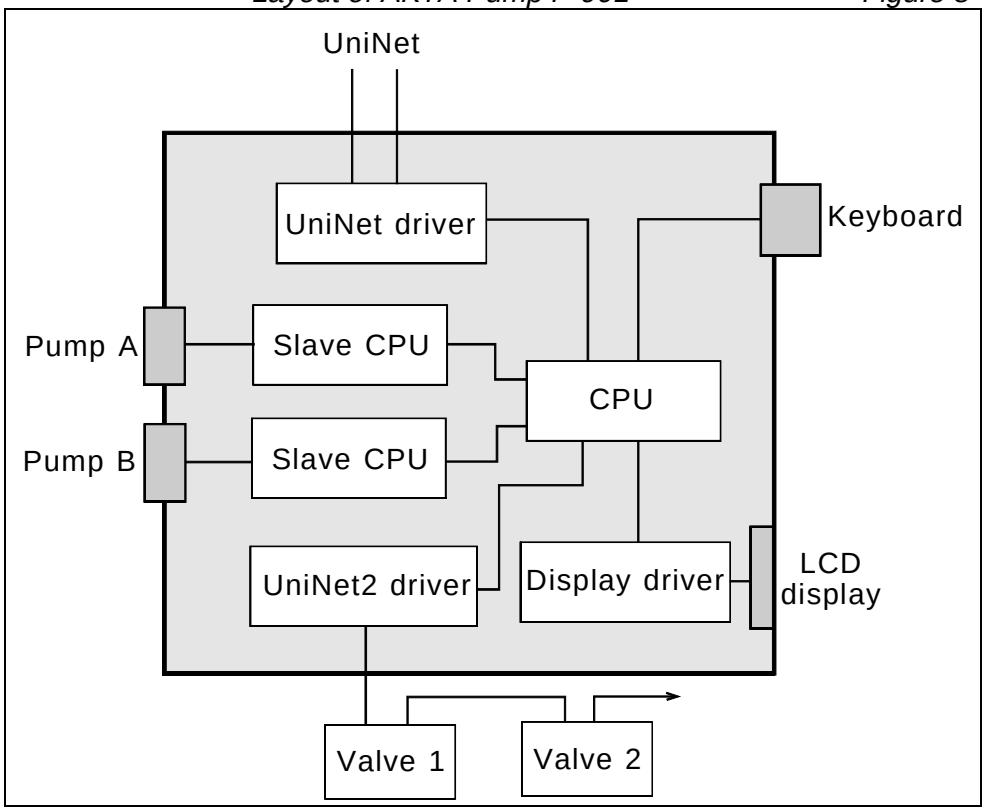
Figure 7



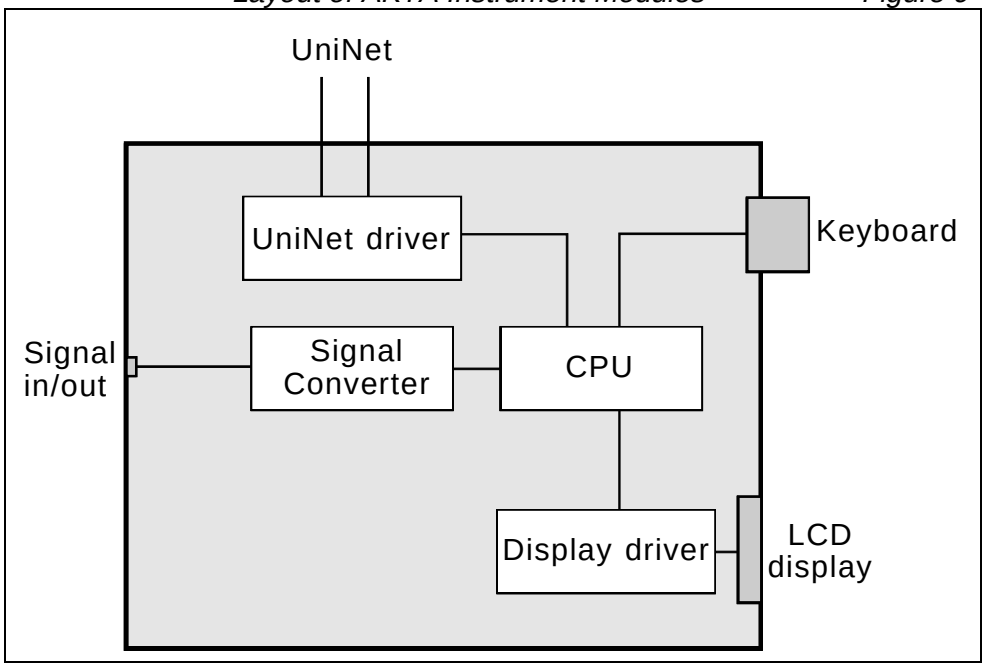
4.3.3. *ÄKTA* Instrument Modules

ÄKTA uses a Controller Board which communicates with the modules through UniNet. Each *ÄKTA* module has its own CPU which allows the module to be used as a stand-alone instrument. Figure 8 describes the functional principle for the *ÄKTA* Pump P-901, and figure 9 illustrates the functional principle for Monitor UV-900, Monitor pH/C-900 and Fraction Collector Frac-900.

Layout of ÄKTA Pump P-901 *Figure 8*



Layout of ÄKTA Instrument Modules *Figure 9*



4.4. Software Design

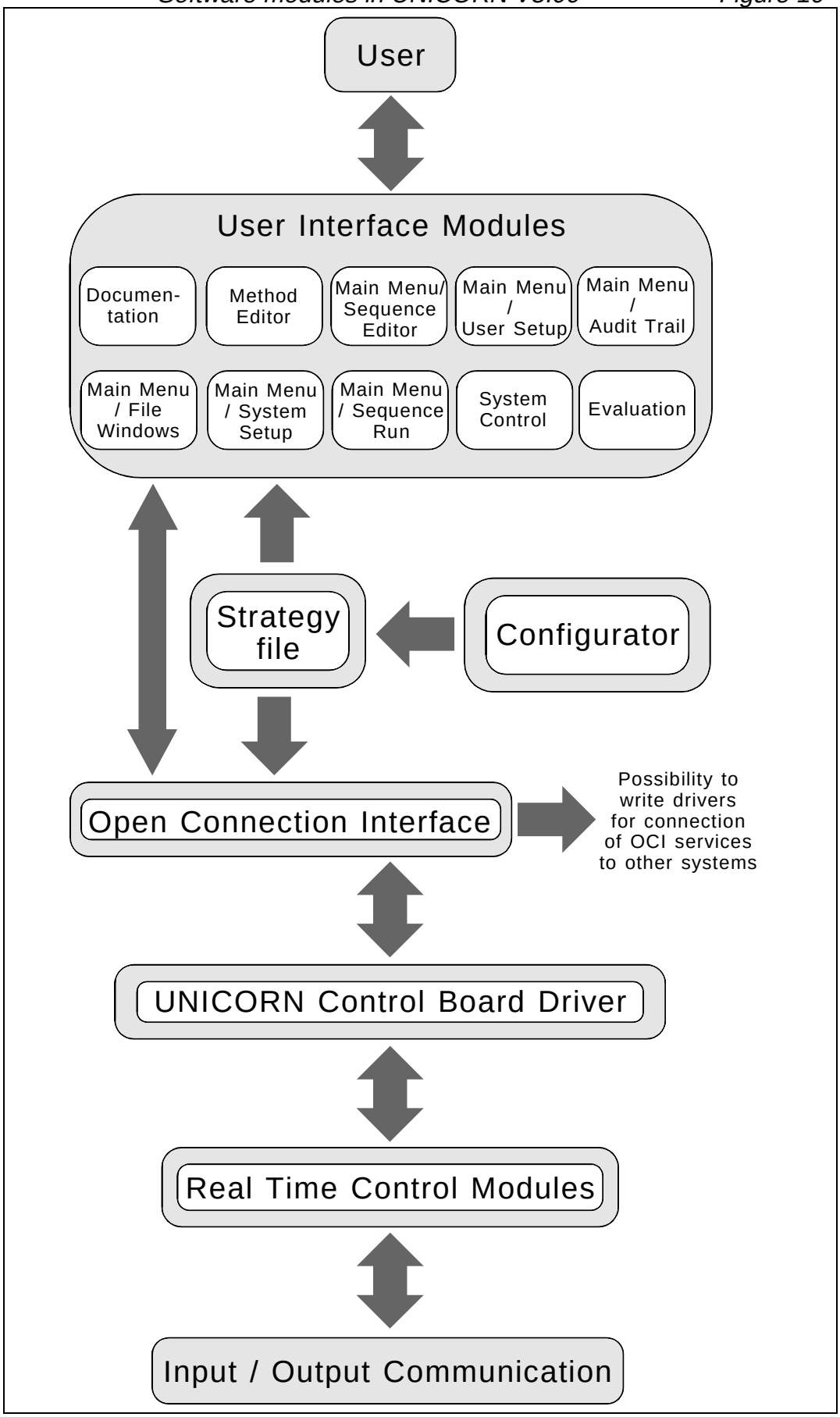
4.4.1. Overview Software

The PC software consists of the following parts :

1. The Configurator is a program used by Amersham Biosciences to configure *UNICORN* to reflect the installed instruments and the flowpaths for each individual system. See also Section 4, Configurator.
2. User Interface Modules that are sold as the product *UNICORN* V3.0N. See also Section 4, User Interface Modules.
3. Control Unit Software, which resides in the EPROM of the Controller Board. See also Section 4, Real Time Control Modules.
4. Open Connection Interface Software, which makes it possible to communicate with other *UNICORN* V3.0N-systems on the same network. See also Section 4, Open Connection Interface.
5. A strategy file, which includes all information needed by the modules Method Editor, System Control, and the Control Unit Software to adapt the software to the systems design and instruments. The Strategy file is defined by Amersham Biosciences using the Configurator program.
6. A program called "*UNICORN* Control Board Driver", which handles the communication between the bus and the Controller Board. This program makes *UNICORN* compatible to different versions of Controller Boards.

The following illustration (Figure 10) on next page shows the program modules and the software structure of *UNICORN* V3.0N.

Software modules in UNICORN V3.00 *Figure 10*



4.4.2. Configurator

The Configurator is a software tool used by Amersham Biosciences to create and define the system unique configuration template file. Examples of what is included in this field :

- Graphical flow scheme and its dynamic behavior.
- Method and Manual instructions available to the user.
- Definition of which signals have “watch” capabilities.
- System Settings available to the user.
- Calibration possibilities available to the user.
- Signals possible to log in Results File and on screen with maximum scan rates of logged signals.
- Run data contents

The configurator definition of a system is saved in a “Strategy file”.

A complete strategy file includes all information needed by the Method editor, System Control, and the Control Unit software in order to conform to the system that has been defined in the strategy.

4.4.3. User Interface Modules

The User Interface Modules (described in detail below) are accessed directly by the user. Entries from the keyboard, selections by the mouse and presentations on the PC monitor are controlled by these modules.

4.4.3.1. Documentation

Documentation is a module that includes information that follows the method through creation, run and evaluation.
Some of this modules contents are:

- Start protocol definition
- Notes
- Result file name definition and location
- Evaluation and report procedure selection
- Reference curve selection
- Questions and answers to the operator upon start of the method
- Template of method instructions that have been defined as variables
- Test scheme and variation of the template parameters over a number of runs
- Method history information
- System settings information
- Calibration and tune parameter information
- Run log information
- Column information
- Buffer Prep information

Documentation is used in the method editor, system control, and by the evaluation module.
Documentation handles the execution of the complete start protocol.

4.4.3.2. Method editor

The method editor is the module employed by the user to define methods. There are several ways to view the method:

- Block window representing the structure of the method.
- Graphical window representing the gradient and flow in a diagram.
- Text window displaying the method in text format.

The flow scheme of the system may also be viewed.
Method instructions are changed, inserted, or deleted using an instruction dialogue window.

When methods are to be run on the CU, the method editor creates loadable methods on request from the system control module.

4.4.3.3. Main menu / Method Queue

The sequence editor is a tool used to define a number of methods to run in Method Queue on the four possible systems that can be assigned to each system control on a *UNICORN* PC.

4.4.3.4. Main Menu / User Setup

In this module users and their access rights are defined, changed, or deleted. All other modules get information on individual user's access rights from this module.

4.4.3.5. Main Menu / Audit Trail

Audit trail gathers information on the use of the system, such as;

- Methods and sequences started and ended.
- Calibrations and tuning done.
- System errors.
- User setup changes.

4.4.3.6. Main Menu / File windows

The file windows displays the contents of the method and result directories that are accessed by the user who is currently logged in.

4.4.3.7. Main Menu / System Setup

In this module, Strategy files and physical systems are assigned.

4.4.3.8. Main Menu / Sequence run

This module handles a running sequence. This includes starting methods according to the defined sequence by using the system control modules on the particular PC.

4.4.3.9. System Control

System Control displays the status of a running system. This includes a dynamic flow scheme, a trend window displaying the collected curves, a run data window displaying selected data in table format, and a logbook with events that have occurred.
Manual interaction capability is available as well as calibration and tuning functionality.

4.4.3.10. Evaluation

The evaluation module includes functions for curve operations, integration and peak table creation, internal and external standard calibration, and report generation.
All used functions can be recorded, thereby defining an evaluation procedure. These procedures can be called from a method or directly in the evaluation module.

4.4.4. Real Time Control Modules

The following software modules reside in the EPROM on the Controller Board. This board is placed in the Control Unit of *BioPilot* systems, and in the system housing of *BioProcess* systems (see also Section 4, Control Unit Description).

4.4.4.1. Communications provider

Handles the communication protocol between the control unit and the PC.

4.4.4.2. Method / Strategy receiver

Builds the Strategy lists and Value table when a strategy is downloaded from the PC.

4.4.4.3. Data handler

Sends packages of trended data to be stored in the result file and/or displayed from the PC.

4.4.4.4. Data Selector

Gets the data from the value table at certain intervals set in the strategy and sends it to the data handler.

4.4.4.5. Manual receiver

Receives manual commands from the PC and sends them to the Method executor module for execution.

4.4.4.6. Event handler

Receives events from all other modules on the CU that are to be sent to the PC. Sends them to the PC.

4.4.4.7. Strategy executor

Calls algorithm blocks with input from and output to the value table periodically, as defined in the strategy.

4.4.4.8. Method executor

Checks method list and manual line up periodically for commands to be executed. All strategy defined instructions are of the same type and execution of these requires writing the parameter values in the value table.

4.4.4.9. Optomux Communication

Handles the periodic update of values in the value table from Optomux devices and handles event-based updates of the value table from the Optomux devices. Sends values when appropriate to Optomux devices.
All according to definitions in the strategy lists.

4.4.4.10. Factory tests

Service tests that can be started from the RS232 communication or from a manual command. Results of tests are reported either to the RS232 communication or through the Event handler module.

4.4.4.11. RS232 communication

Handles the communication over RS232 used for factory tests (and possibly service).

4.4.5. Open Connection Interface (OCI)

The open connection interface module executes independently of the user interface modules.
It handles all data relevant to the ongoing run and communicated through the drivers with the control unit.
User interface modules have access to a number of OCI requests to get the data they need and to influence the connected system. This interface is implemented using named pipes.

The OCI interface used by the User Interface Modules may also be used by drivers to other software systems such as LIMS, plant management systems, DDE connection, etc. Such drivers are, however, NOT included in the base *UNICORN* software.

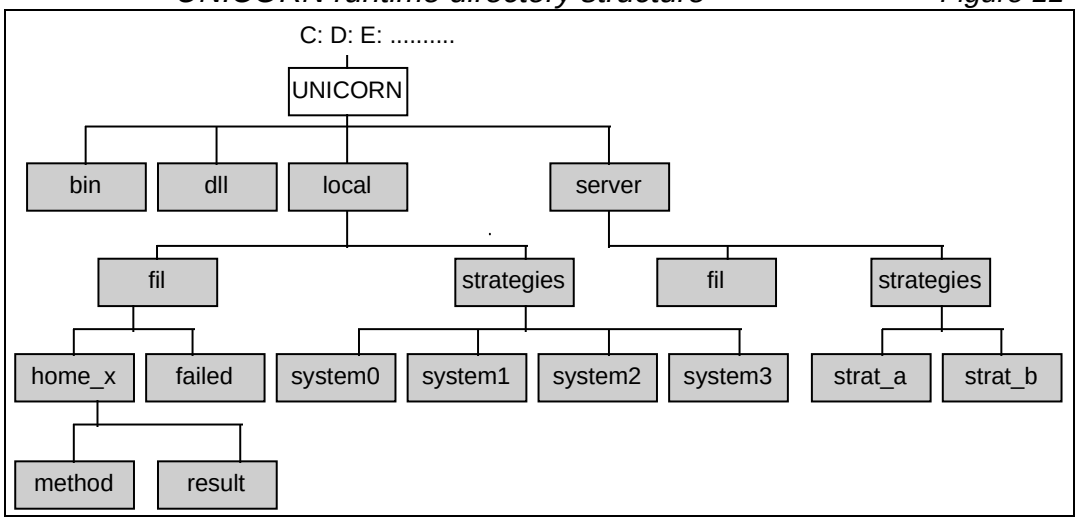
4.4.6. *UNICORN* Directory and File Structure

UNICORN V3.0N has a global file structure as shown in Figure 11 (below). However, since *UNICORN* V3.0N can run both in a netware environment and as a single local station, some differences are found in the directory structure.

The subdirectory “Server” will exist on the shared hard disk (if *UNICORN* is installed in a netware) or on the local hard disk.

The subdirectory “Local” will exist on the local hard disk and hard disks with created “Homes”.

UNICORN runtime directory structure *Figure 11*



The following table explains the functions of the different directories in *UNICORN*.

Directory	Description
C: D: E: ...	Unicorn is installed on ONE drive but multiple LOCAL directories may reside on several drives simultaneously
unicorn	The main Unicorn directory.
bin	The executable modules.
dll	The Unicorn dll modules and device drivers.
local	Holds a copy of the original system description table.
fil	Hold a copy of user file.
home_x	Home directories, holds methods and result file directories. A user is always assigned access to minimum one (the default) home directory, but may also be assigned access to several home directories.
method	Holds method files, sequence directories and user created subdirectories
user_x	User created subdirectories for methods.
result	Holds result files, scouting runs and user created subdirectories.
user_x	User created subdirectories for results.
failed	Subdirectory for results if network is lost.
result	Current results which where not copied successful after method end
strategies	Holds subdirectories for each installed system.
system0	System information for system0. Holds a copy of the latest strategy, picture files and templates. Holds the original system setting file and system audit file.
system1	as above but for system 1
system2	as above but for system 2
system3	as above but for system 3
server	This directory handles Unicorn in a netware. It's located on the server if Unicorn is installed with netware otherwise on the local disk. Holds the original file of system description table (system.tab)
fil	Holds the original file of users (users20.mpm). Holds the global user audit files.(*.slg) Holds also the files for evaluation procedure library,colonn list and receipt list (eval.res, colonn.nam, receipt.rcp)
strategies	Holds a directory structure of all available strategies.
strat_x	Sub directories for a strategy. Containing strategy, picture files, templates and default system setting.

5. SOFTWARE QUALITY ASSURANCE DOCUMENTATION - *UNICORN* V3.00

5.1. Intention

This section describes the quality assurance documentation obtained in the process of developing *UNICORN* V3.00.
The intention is to explain and indicate how this was performed and documented, and not to provide an all-inclusive documentation package.
The complete documentation is available for audits at Amersham Biosciences in Uppsala, Sweden.
New *UNICORN* versions are developed in projects. All documentation created within the project uses this project number as a unique identifier.

5.2. Software Quality Assurance Documentation

UNICORN V3.00 was developed in accordance with the principles described in Section 3. The names of the parts in this section refer to those stated in Section 3, System Maintenance and Update Policy.

5.2.1. Technical Specification (TS)

The desired functions for *UNICORN* V3.00 were specified in the Technical Specification (TS). Observe that the TS only specifies required changes needed in order to obtain the product, which in this case is *UNICORN* V3.00. New versions of *UNICORN* are developed in projects with unique project numbers for each new version, which is used on the front pages of all resulting documents. The project number for *UNICORN* V3.00 is specified in the TS.
Observe that the TS does not describe how the modifications are to be performed, but which functionality is desired in the new program version.
Below is a copy of the Table of Contents from the TS for *UNICORN* V3.00.

When the desired functional changes (as described in the TS), involve components other than software, a Technical Specification for these parts is required. The development of *UNICORN* V3.00 addresses only software; therefore, the Technical Specification (System) and the Technical Specification (Software) are the same document.

5.2.2. System Design Description (SDD)

The following text is quoted from ”General” in the SDD”, and it describes the intention of the document:

> ”The purpose of this document is to describe the design of the software *UNICORN* version 3.00.
> A modularization on a high level is described
> The communication principle between *UNICORN* 3.00 controller and instruments is described.
> For detailed design descriptions per module refer to the Module Design Descriptions.”

This following part is a copy of the Table of Contents of the SDD for *UNICORN* V3.00 :

5.2.3. Module Design Descriptions (MDDs)

UNICORN software consists of modules which are described in Module Design Descriptions (MDDs). Each MDD may have submodules which describe in detail its functionality, design and performance.

There are two reasons why the whole Module Design Description (MDD) is not described in detail in this document :

A detailed description of the MDDs would reveal structural information about *UNICORN* V3.00. This is information Amersham Biosciences wishes to keep from competitors. This is probably of a more confidential information nature than the source code.

The total documentation consists of more than 130 written documents describing the MDDs in V3.00 of *UNICORN*. It is not possible to print its contents or describe it in detail here.

The following is a copy of the Table of Contents of an MDD used in the development of *UNICORN* V3.00.

5.2.4. Module Test Plans (MTPs)

The module test plans are individually constructed to verify the design and performance of each module. The individual tests are constructed based on an expected result, and then compared to this result. They also describe a procedure for the person carrying out each Module Test. Each

MDD has a corresponding MTP. Below is an example the Table of Contents for the Documentation Module.

5.2.5. Module Test Reports (MTRs)

Module Test Reports are the documents based on the MTPs specifying the results from the tests, who has performed the tests, when, and the compliance with expected results.
The MTPs were subject to individual approval before the modules were integrated with the other parts of the program, in accordance with the principles described in Section 3, System Maintenance and Update Policy.
Any deviations from the expected results are reported and evaluated for each module prior to integration with other modules.
In most cases, the different tests (as described in the respective MTP) are “ticked-off” and signed. The completed “tick-off”-forms and other types of protocols with obtained results are included in the MTRs.

As an example of a module used earlier in this document, the MTR for the “Documentation”-module has the following contents:

References which refer to the corresponding MTP and SOPs used when the module was tested.
Appendices containing all obtained “tick-off” forms and other protocols from the tests performed.
Summary containing the evaluation of the test, and a statement that this module may be integrated with the other modules in *UNICORN* V3.00

Deviations specifying that one minor deviation was found during the test. It describes where and how this deviation was triggered.
Material specifying which computer (make, model, RAM-memory size, and free hard disk space),and operating system that were used for the tests.

5.2.6. Integration Test Plan (ITP)

The Integration Test Plan is used to verify that all modules work together as intended.
The ITP for *UNICORN* V3.00 describes how the modules shall be tested together. It has the following contents (Table of Contents copied from the original document) :

5.2.7. Integration Tests Report (ITR)

The Integration Test Report is based on the same document as the ITP, but also contains the following :

a) A summary of the results from the test. The following table is a copy of the Table of Contents from the ITR.

If any deviations had been found, a unique Deviation Report number had been denoted on the ITR, referring to each specific deviation.
b) Tested functions which meet the specified demands are “ticked-off”.
c) Description of conditions and instruments used in the Integration tests.

5.2.8. Software Test Plan (STP)

The Software Test Plan is intended to define the criteria and tests for *UNICORN* V3.00. The STP does addresses the software, not the system with instruments and other devices. The system (with instruments and devices) is addressed in the SVP.

The following is a copy of the *UNICORN* V3.00 STP contents :

4. TEST PROCEDURES 3

5.2.9. Software Test Report (STR)

The Software Test Report is a document in which the different deviations are specified, and in case of software errors, are measured and classified. The frequency for software error occurrence is measured under different conditions, and compared to previous versions of *UNICORN*.

This is the contents of the STR for *UNICORN* V3.00 :

References to applicable SOPs and the STR.
Appendices to the STP and a list of corrected and verified errors.
Summary the weighted error frequency obtained is stated.
Error frequency contains the data obtained (in some cases graphically), and defines and classifies the error types.
STP application is a matrix with different *UNICORN* versions and the STP references.
Test Resources tells how many people have been working with this test and the time spent for the test.
Error Classification defines the types of errors in different occurrence classes.
Test Environment describes the environment used for the tests (OS, computers, file system types, communication boards, netware OS and printer/plotters).

5.2.10. System Verification Plan (SVP)

The System Verification Plan (and the PTR) has similar construction as the STP, but addresses the whole chromatography system and not only software.

The SVP describes the tests to be performed and criteria for each one of these. Every test has two boxes marked ”T” (for “tick-off”, if approved) and ”D” for deviation. In the case of a deviation in the obtained result, the Deviation Report number is denoted in the ”D”-box.

5.2.11. System Verification Report (SVR)

The SVR for *UNICORN* V3.00 describes the result obtained when the SVP was performed. It specifies computers and chromatography systems used during the testing.
The error frequency is compared to previous versions of *UNICORN*.
The errors has been classified in the following categories :
System Stops, Functional Errors, Aesthetic Errors, and Suggestions.

5.2.12. Market Readiness Review (MRR)

The MRR (as described in Section 3, System Maintenance and Update Policy) is intended to verify that all considerations for the release of a new product (in this case *UNICORN* V3.00), have been made.
The MRR for *UNICORN* V3.00 is based on form 70-7600-43 ed. AG and consists of a main document with approval of the MRR for the project and references to reports verifying that the required activities have been performed.
These are the sections referring to *UNICORN* in the main document :

1. **Marketing Specification**
   Verification that *UNICORN* V3.00 conforms to the marketing specification, this is also verified with regional companies.
2. ***This section contains confidential information, which has little or no relevance for validation efforts.***
3. **Product information**
   Verifies that Instruction Manual, Service and Maintenance manuals have been produced and are available.

4. ***This section contains confidential information, which has little or no relevance for validation efforts***
5. **Existence of adequate organization for distribution and service.**
6. **Training finished/planned at subsidiaries and Product Division.**
   Verifies that specialists, sales representatives and service engineers have or, will receive adequate information and training.
7. **Field testing, alfa and beta seed sites. Results, action plan.**
8. **Verification testing etc. according to QA procedures.**
   Verifies that the following has been performed:
   Quality audit of project reported.
   Design Reviews: actions, action plans.
   Verification testing completed.
   National and international standards and regulations complied with, including certification testing.
   General requirements and guidelines for product safety complied with (Amersham Biosciences Standard no 70-0200-97, 99, 99E and MSDS).
9. **Producibility.**
   Verifies that the specified number of copies of the program have been produced, and the production facility is capable of producing adequate numbers of copies.
10. ***This section contains confidential information, which has little or no relevance for validation efforts.***

5.3. Availability of documentation and Source Code

The Source Code can not be included in this documentation. However, if a customer has purchased a license to use *UNICORN* and needs to validate the software for FDA or corresponding authorities, this can be accomplished through a Source Code Escrow Agreement combined with a Confidential Disclosure Agreement with the customer and Amersham Biosciences.

The SOPs used and documentation obtained during the development and approval process of *UNICORN* is available for customer audits at Amersham Biosciences in Uppsala, Sweden.

5.4. Project Personnel Credentials

Amersham Biosciences employs and uses only skilled programmers and project personnel having University degrees in computer technology or programming, or having similar education or experience.

5.5. User Instruction Manuals Review

The basic material used in the development of the manual for *UNICORN* V3.00 was the manual from *UNICORN* V2.30. The material was reconstructed and / or changed to reflect the functionality of *UNICORN* V3.00**.**

5.6. Problem and Bug Policy

All software contains bugs and / or has malfunctions due to programming errors or other problems. This is also the case with *UNICORN* V3.00, which contains a few bugs and problems which have been discovered at Amersham Biosciences after release, or have been brought to our attention by customers.
Amersham Biosciences wishes to inform all customers using *UNICORN* of all known bugs and problems which have been discovered, knowing that the performance of *UNICORN* may affect the customers applications. Amersham Biosciences considers this vital information for all customers using *UNICORN* as control program.
For this reason detailed list which describes all known bugs and problems and suggested actions to avoid their occurrence are provided with *UNICORN*.
See also Section 3, page 19.

6. VALIDATION OF A *UNICORN* BASED CHROMATOGRAPHY SYSTEM

6.1. Introduction

Validation is a process in which the system owner (who is responsible for the system) performs actions to determine that a system and process, are consistently capable of producing a product which meets its pre-determined specifications and quality attributes.
These actions should involve Design and Requirements analyze, Installation Qualification, Operation Qualification, Method Development and Verification, Education, and other actions that the system owner finds relevant in order to allow the system to start. The validation process should end with an approval from the system owner of all previous activities, meaning that a high degree of assurance which ensures that the system is fit for its intended use has been obtained.
After the system owner has approved the system to start production (or other intended activity), the system can be called validated.
This means that the word “validated” does not specify any particular level of validation, it simply means that the system owner has approved of its fitness for its intended use.
Validation of a system should not be performed simply to please authorities, but to obtain a system which operates as intended. Regulations emphasize that the system should not produce harmful or subquality products.
The following document contains a suggested validation strategy for a *ÄKTA*, *BioPilot* or *BioProcess* system. Following the procedures explicitly will not give any guarantee that a system which complies with official regulations is obtained.
However, if the procedures suggested here are performed, completed and modified as needed, a validated system can be obtained.
We strongly advise you to be well informed of the regulatory aspects that are related to your product and process. From a regulatory point of view, the system owner is always responsible for the performance of his/her system, and an established opinion of inspectors is that if written evidence of an activity cannot be presented, the activity has not been performed.

These recommendations are mainly based on the U.S. FDA’s Good Manufacturing Practice (GMP), since they are widely accepted throughout the world. Section GMP References in this chapter, page 83, contains references related to GMPs.

The intention of this document is to give the persons involved in the validation of *ÄKTA*, *BioPilot* or *BioProcess* systems advises and a brief guidance and procedures that may be of interest.

6.1.1. *ÄKTA*, *BioPilot* and *BioProcess* systems

Validating a *BioPilot* , *BioProcess* or an *ÄKTA* system can be performed in accordance with the following illustration (Figure 12)

Validation order *Figure 12*

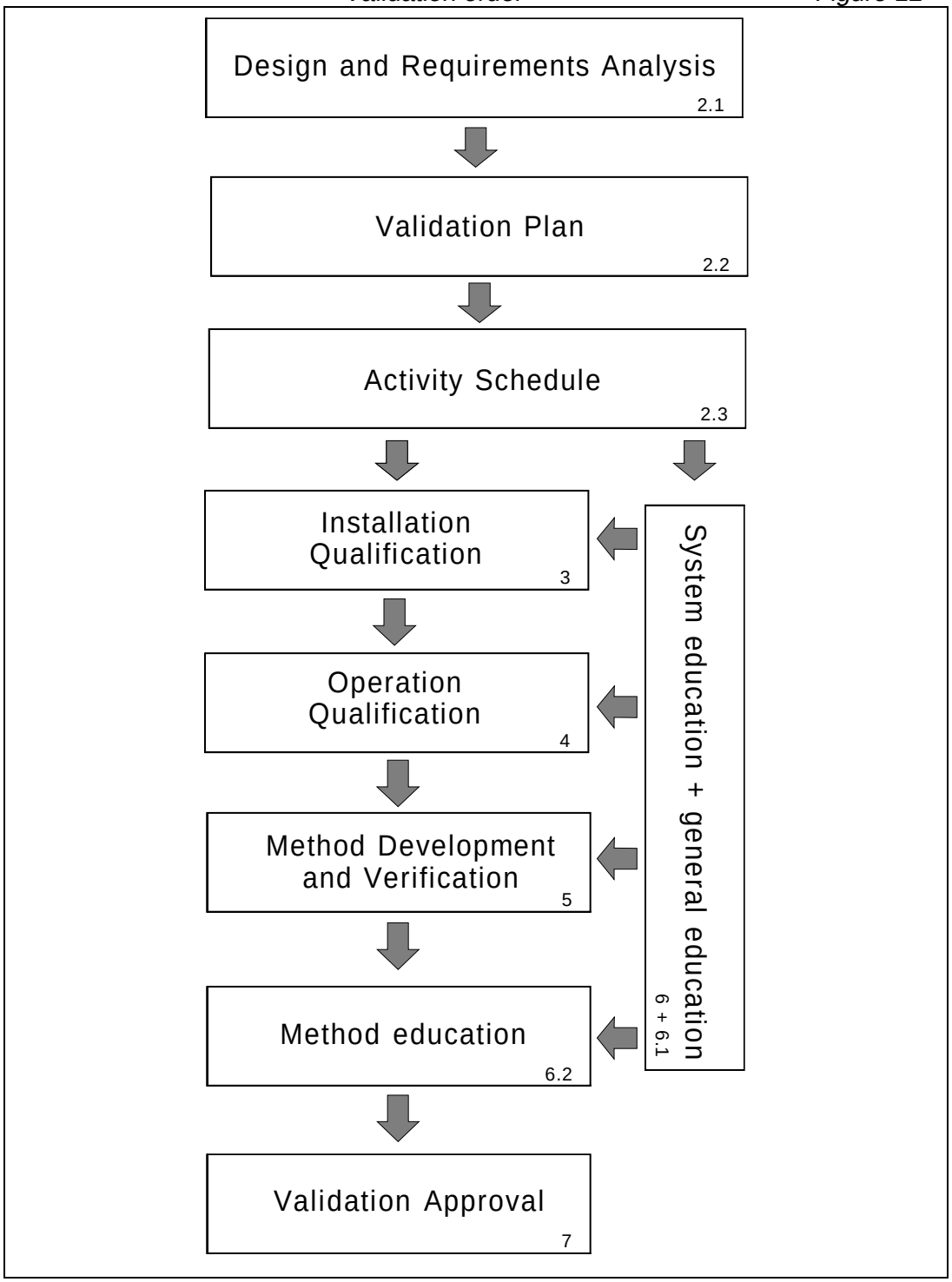


Figure 12 illustrates that validation is a process with components, and not just a task which has to be carried out before a system is put into production.

6.2. Design, Validation Plan and Activity Schedule

6.2.1. Design and Requirements analysis

System design is vital to its function. Amersham Biosciences provides design assistance to its customers and most systems delivered are designed in accordance with customer demands.
The design phase should not only involve the construction of the chromatography unit, but also take into consideration the intended use of the system. The intended use will dictate the required level of validation. For example, the requirement for production of a therapeutic product is quite stringent.

6.2.2. Validation Plan

A Validation Plan should include all components required to obtain the desired validation level. The plan and specified activities are fundamental in the validation of the system, and should be approved by the system owner.
The Validation Plan should include all tests and verifications needed to obtain the desired validation level. This includes not only items mentioned here, but also other relevant activities, e.g., supplier audits. The earlier chapters of this documentation describe the development SOPs and procedures Amersham Biosciences uses to develop and supply *UNICORN*. This information is part of a validation procedure of a *BioProcess*, *BioPilot* or an *ÄKTA* system, and can be included in the owners validation documentation.
As mentioned earlier, tests of the system during its operation, will almost certainly not be all inclusive. This implies that the design of a system is of great importance.

6.2.3. Activity Schedule

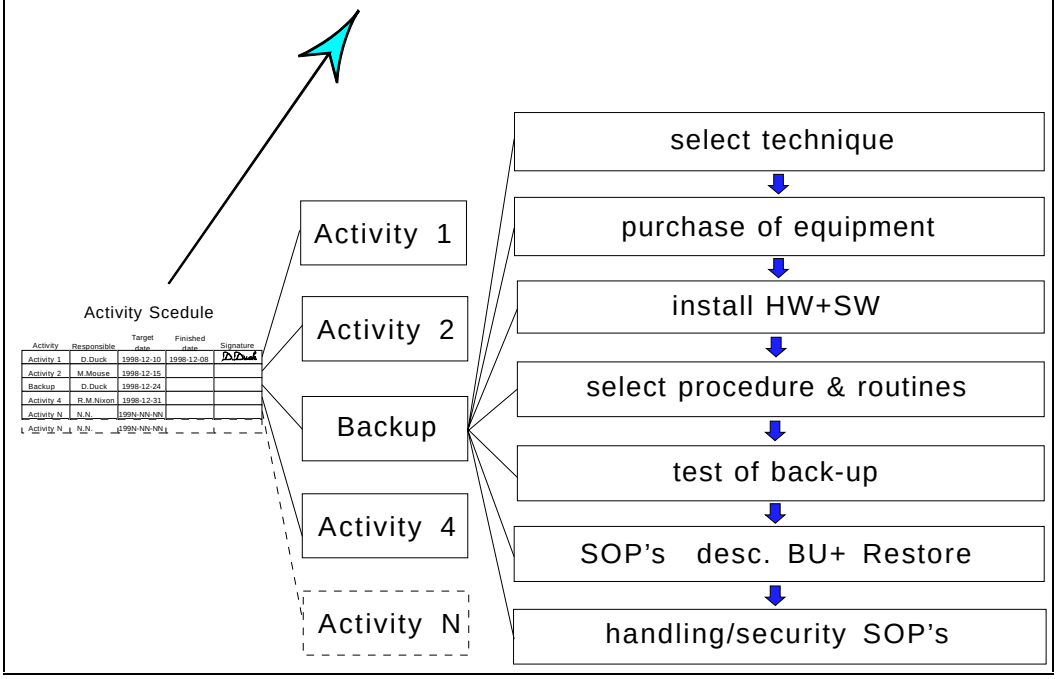
From the Validation Plan, an Activity Schedule can be extracted, specifying each activity and responsible person(s). The activity schedule can be constructed in different ways, but it is important to recognize that activities must preceed other.

A few activities, however, can be performed independently of others (e.g., system and general education).
General education is dependent mainly on two factors that the personnel involved in the development and the handling of the system should have been selected, and the education should be performed prior to use of the system in production.
System education can be divided in two major parts; system specific education and handling education, the latter being dependent on the existence of an operative system. System specific education can be performed during the IQ and OQ of the system.

Figure 13 illustrates an example of an Activity Schedule with sub-levels, describing the development of a backup system.

Activity Schedule *Figure 13*

Activity	Responsible	Target date	Finished date	Signature
Activity 1	D.F. Duck	1998-12-10	1998-12-08	D.Duck
Activity 2	M.Mouse	1998-12-15		
Backup	D.Duck	1998-12-24		
Activity 4	S. Laurel	1998-12-31		
Activity N	O. Hardy	199N-NN-NN		
Activity N	N.N.	199N-NN-NN		



The **Activity Schedule** should in this case contain information describing who is responsible for the development of a backup procedure and a date when this activity should be finished. It should also refer to the "Backup"-activity document (below), in which more detailed information can be found.

The "**Backup**"-activity document should describe the specific steps (and responsible persons for these), needed in to obtain a complete and documented routine. It should also include an official approval, when completed, from the person responsible for the "Backup"-activity.

"**Select technique**" as a suggested first step, would be to determine which different backup techniques are available, and which one of these is preferable, considering security, volume, economic, technical aspects and the effects of a data loss. These considerations should of course be made in conjunction with the system owners' expectations of the system.

Suggested documentation : A document describing the investigation considering all aspects involved.

"**Purchase of equipment**", refers to the ordering and delivery of the necessary hard and software.

Suggested documentation : Notes concerning which equipment was ordered, by whom and when. Delivery note(s) for the equipment (including hardware(HW) and software(SW)).

"**Install HW and SW**", should describe how, were, when and by whom the backup system was installed and connected to the system.

Suggested documentation : A document describing how the installation was performed, which directory holds the software, the files in this directory, function of the files, and how the backup device is connected to the computer.

"**Select procedure & routines**" refers to two things:

1.To decide the frequency and extent of the backup procedure. If the system owner has decided that system should meet high demands in these aspects, a procedure were generations of backups are created should be established.

2. Decide back-up and restore routines that will meet the security demands the system owner has for the system.
The routines should involve issues as: handling of backup media, storage, accessibility and who decides when (under which circumstances) a restore shall be performed , how frequently the backup media should be replaced, how the media should be labeled, responsibility for the routines, and other relevant considerations.
Suggested documentation: A document describing key issues, and why and by whom routines and functions were chosen.

"**Test of backup**" should verify that the routine works as intended, and that the contents of a back-up can be restored. Observe that the whole routine should be tested, not only that the files included in the backup can be restored.
Apart from the knowledge that a back-up routine works as intended, and the contents of the hard disk can be restored, the test of back-up would also provide knowledge of any obstacles that may occur during the procedure, and the time it takes to perform a restore to another computer.

Suggested Documentation : A protocol describing how the test was performed with results and evaluation of the test.

"**SOPs describing backup and restore**". Refers to the creation of the official documents describing in detail exactly how backup and restore routines are performed.

Suggested Documentation :
1 - A Back-up SOP, describing in detail exactly how a complete backup is performed.

2 - A Restore SOP, describing how the contents of a media containing a backup are restored partially or totally.

"**Handling/security SOPs**" are the instructions describing how and where backup media should be handled and stored.

Suggested Documentation : SOPs regarding:
- responsibilities
- frequency for backup, generation handling
- handling of media containing backup, storage and accessibility
- who has the authority to order the use of a backup
- when backup media is "worn out" and should be replaced
- person(s) in the system responsible for the backup/restore
- changes in backup routines
- log for back-ups, labeling of media (e.g., tapes)

6.3. Installation Qualification (IQ)

Installation Qualification* is defined by the PDA as "documented verification that all important aspects of installing the hardware and software adhere to the computerized system specification. Installation qualification also includes verification that there are appropriate manuals, as-built drawings, instrument calibration reports, and SOPs on the operation and maintenance of the system. Installation qualification of software includes verifying that the proper version of the program has been installed and that the appropriate back-up copies exist."
The Installation Qualification verifies that the delivered system is complete, and verifies all components in the system meet specified requirements. All components are described in such detail that a similar complete system can be built with the information from the IQ.
We do not recommend customers assemble IQ specifications and protocols. The reason for this is that a customer will not know all details and criteria needed for a complete IQ e.g., jumpers, interrupt addresses, files required etc.
Amersham Biosciences has a unique knowledge of all components in the system, and has developed a complete package for this purpose. Much time and effort can be spared by using the Amersham Biosciences Fast Trak Validation Services. More details describing these services are described in Section 6, IQ and OQ Services from Amersham Biosciences Fast Trak Validation.

6.4. Operational Qualification (OQ)

The PDA has defined operational qualification* of a computerized system as : “documented verification that the system operates in accordance with the computerized systems specification throughout all anticipated operating ranges. Operational qualification may be performed on the integrated system or on each subsystem and includes the identification of all important operating parameters, their anticipated ranges, appropriate acceptance criteria, and the tests to be performed to demonstrate that the system meets the acceptance criteria. Operational qualification also includes performing specified tests and reporting the results.”
Amersham Biosciences Fast Trak Validation offers extensive services in this task, described in Section 6, IQ and OQ Services from Amersham Biosciences Fast Trak Validation, which will provide a customer with test protocols and documentation proving that the system is capable of performing its required functions.

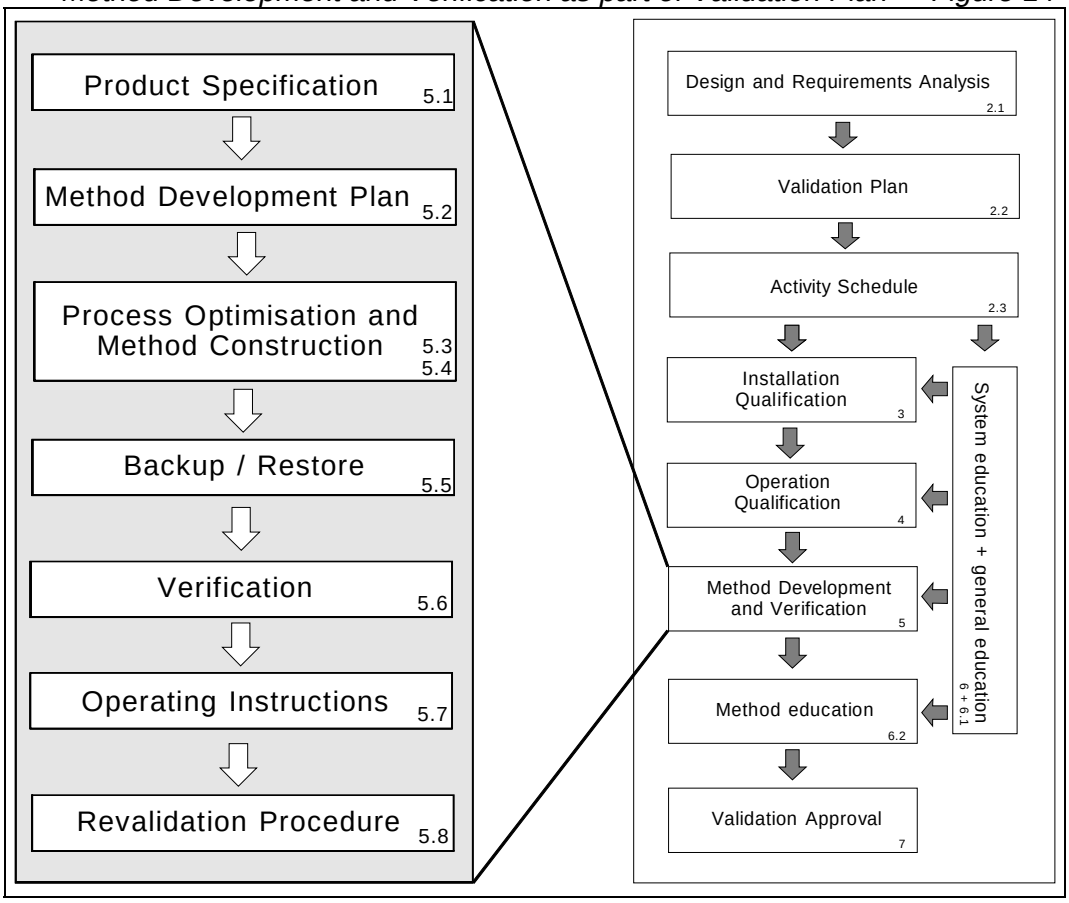
*see Section 7, page 84 for references.

6.5. Method Development and Verification

The development of methods are dependent on many factors as the system is complex and allows the user to freely construct and make modifications as desired. Method development is of great importance to the demands you may have on documentation, security levels, and operating procedures.
An *FPLC* system used for scientific research normally has much lower requirements in these aspects than a *BioProcess* systems which is used under FDA regulations to produce pharmaceuticals. The level you choose is highly dependent on the systems use.
It may appear unnecessary to apply this way of developing systems in many cases, such as R&D activities; however, it is common that processes are developed on an *FPLC* or *BioPilo*t and scaled up to a *BioProcess* system at a later stage. If this is the case, structured development and good documentation will be beneficial to the scaling up and documentation of the process at a later stage.
It is recommended that the system use is determined first, and then the required level of quality. These two factors should be the fundamentals for the development of your own methods.
Keep in mind that you never can test quality into your system - quality begins at design level.
We suggest (as earlier mentioned) that the system is handled in accordance with PMA's "System Development Life Cycle" when your own programs and methods are developed.
Amersham Biosciences Fast Trak Validation offers a large number of services to support customers in these matters.
To verify that a developed method works as intended, we suggest that you start with a Product Specification describing the aim of the process. After this, a Method Development Plan should be established and include details of the activities needed to fulfill the Product Specification.
The Process Optimization and Method Construction are integrated activities intended to develop a method which chemically fulfills the Product Specification.
The procedure should continue with a Verification in which the earlier activities are verified and approved.

The last steps in the Method Development and Verification is to establish Operating Instructions and a Revalidation Procedure. The Operating Instructions should be operating procedures for every function in the system and should describe the responsibilities for each function of the system (who is responsible for what).
The Revalidation Procedure should describe what should be controlled, verified and tested regularly, and the criteria for the approval of a revalidation.
This can graphically be presented as shown in Figure 14.

Method Development and Verification as part of Validation Plan Figure 14



6.5.1. Product Specification

The first step in the development of a new process is to determine process goals. Requirements for product purity, activity, and quantity should be specified. Economic considerations can also be included in the Product Specification.

6.5.2. Method Development Plan

The activities needed to obtain a method which complies with the Product Specification should be specified in the Method Development Plan. The plan should also describe which documentation should be result from the activities.
This plan should preferably be organized in a structured manner. Responsibilities for each activity should be stated.

6.5.3. Process Optimization

Developing a process to optimize yield is often a complex and complicated task. Amersham Biosciences Fast Trak offers expert services in process optimization. The main goal of process optimization is to develop a method that produces a product of a desired purity and quantity.
Process Optimization includes several steps:

- Raw material considerations.
- Specification of the process.
- Choice of techniques.
- Number of chromatographic steps.
- Yield in each step.
- Method scouting.
- Method optimization.
- Economic considerations.
- Scale up and adjustments.
- Media life length.
- Cleaning in place (CIP).
- System sanitization.
- Other activities relevant for the development of the individual process.

These activities will generate documents which will be beneficial to the verification of the process.
Accordingly, all relevant documentation should be created, and properly stored.

6.5.4. Method Construction

As there are many aspects of how a program should be constructed and how programming should be performed, no recommendations are suggested here. Regardless of the method used for programming, documentation should be created to support and verify the function of a program.

Examples of such documents are:

- Functionality model.
- A list of all variables and description of their function.
- Limits for variables and fields.
- Method instructions.
- Description of all modules and subroutines in the program and their functions.
- Signatures and dates of the responsible programmer(s).
- Descriptions of all alarms and warnings.
- Protocols from tests performed by the programmer(s).
- Security and access levels.
- Other relevant information.

When a program has been constructed by the programmer(s), it should be subject to testing and evaluation in order to be approved for use as intended.
The testing procedures are dependent on the demands and validation level intended for the system, and cannot, therefore, be specified here. The test should not only prove that the program performs as intended, but should also include "worst case" simulations in which warnings and alarms are triggered. These tests should not be carried out or constructed by the person(s) who constructed the program.

Suggested documentation :

- Test plan.
- Test description.
- Responsible and responsibilities.
- Criteria for approval.
- Protocol from test.
- Criteria for approval of test.
- Other relevant documentation.

6.5.5. Backup / Restore

Backup and restore are preventive actions to avoid accidental data loss from the system. These actions assure that if data were lost from the computer's hard disk, they could be restored. Another (new) hard disk is normally used for the restoration.
It is almost impossible to predict when and how a computer's hard disk might malfunction. Hard disks are partly mechanical devices, and do have a limited lifetime.
It is important that these issues are dealt with prior to the start of production. A functioning backup system is one of the validation demands the system owner should have in order to approve of the system's to be use in production.
Different media (storing techniques) for the contents of the computer's hard disk can be used, but magnetic tapes in cassettes are the most common. Magnetic tapes are inexpensive, have a large storing capacity, and can be overwritten with new data when so desired. One disadvantage is that they are quite time-consuming to use, therefore many systems use automatic routines which perform backups at night.
When a backup routine is developed, generations of backups should be considered, since not all problems that may occur in a system (and originate from the hard disk) are discovered instantly.
If magnetic tapes are used, they should be replaced at timely intervals, according to the supplier's specification.
Backup routines should be tested in order to verify that the desired files really are restored from the media when the restore routine is used. Verification of this should be included in the validation of the system.
Storage and handling of the media containing backups, should also be considered when a backup routine is constructed. To prevent simultaneous loss of both backup media and contents on the computers hard drive, they should be stored in different locations.

When designing routines for the backup, remember that the media with backups may hold vital confidential information about your production process.

Operating procedures describing how a restore is performed should be developed and tested. There should also be documents stating the responsibilities for backup and restore. For the restore routine, it is important to state which company function that has the responsibility for deciding when and under which circumstances a restore shall be performed.

An example of how a backup/restore routine can developed is described under Activity Schedule in this section 6.

6.5.6. Verification

As mentioned earlier, the performance of the system is highly dependent on its design, not only the fluid handling side and the hardware, but also the programmable part of the computer program. The program should, therefore, be constructed with a functionality model that minimizes the risk of errors caused by poor design.
When all the tests of the computer program and the chromatographic process have been performed (according to the Method Development Plan) and documented, the method should be approved by the person designated to be responsible for the system.

6.5.7. Operating Instructions

The Operating Instructions are an important part of the validation process. Their main function can be described as "instructions in how to handle the system when in operation". In the "System Development Life Cycle", the chapter entitled "Operation and Maintenance" addresses these questions and specifies what should be considered.
In cases where the system is intended for production regulated by official rules, these recommendations should be modified accordingly.
Observe that these recommendations are not “batch specific”, which means that they do not specify what procedures, documentation, etc. are required for such production.
The Operating Instructions should contain the following instructions and documentation :

- Routines for backup and restore.
- Adding / modifying / deleting users.
- Responsibilities for the system.
- Functions in the system.
- Organization around the system.
- Security (routines and responsibilities). Change control procedures.
- Log book for the system.
- External contacts and companies (suppliers).
- User education.
- User manuals.
- Frequency of revalidation.

- System maintenance (schedule and procedures).
- Cleaning instructions.
- Calibration schedule and procedures.
- Other relevant SOPs for the operation of the system.

The Operating Instructions are a part of the validation and specify how the system should be handled when in operation. The Operating Instructions should be updated whenever any system related changes are made. Such changes shall also be recorder, dated, and signed in the system log book, in accordance with the Change Control Procedures.
An important part of the revalidation is to verify that the Operating Instructions are updated and handled correctly.

6.5.8. Revalidation

The U.S. FDA recommends that all computerized systems are frequently revalidated and that there is a specific revalidation procedure for each system. Since there is no recommended frequency for revalidation this has to be decided by the system owner.
Revalidation does not mean that the system has to be validated all over again, but rather that there should be a specific procedure which ensures the system owner that the system works as intended after a specific period of time.
Accordingly, it is recommended that the system owner has a Standard Operating Procedure which describes the frequency and protocols for revalidation.
Observe that calibrations (which should be carried out frequently) are not revalidation.
Wherever applicable, the revalidation procedure should contain specific test data sets and procedures, which when processed in the system should give specified results that are to be compared to those expected.
A suggested revalidation procedure cannot be specified here since it is site-dependent, however; the following items should be included :

Verification of the following documentation;
- Log book.
- Updated user manuals and SOPs for all activities.
- Documents from the system validation.

Verification that the following has been carried out in accordance with SOP's:
- System changes (according to change control procedures)
- Log book entries.
- Security procedures.
- Calibrations.
  Sanitizations.
- Errors and disturbances reports.
- Education.
- Service and maintenance.
- Backup.
- Adding/deleting users.

System test procedure (data for these tests should be attached to the revalidation procedure)

- Component tests.
- Program test.
- Total system test.
- Performance test.

Additional items should be included if they are relevant to the functionality of the system.

6.6. Education

Each user and operator should be educated for his/her specific function. This is not only an regulatory demand, but a way to ensure a system that runs smoothly with a minimum of errors. Most of the mishandling errors are caused by inexperienced users. Education is, of course, especially important for the person(s) responsible for handling the operating system, Windows NT. Mistakes in the operating system can lead to all types of disturbances in the functioning of the system. The operating system has, by necessity, many powerful commands that can be fatal to the system if exercised without the knowledge of their effects on *UNICORN*. This suggests that the person(s) responsible for the operating system should have sufficient and updated knowledge in its handling, and that its use should be minimized. It is also important that the knowledge is current. Education should take place shortly before production startup of the system. If the system is to be used to produce GMP-controlled substances, general GMP education is also required.
If the system is modified, users should be informed and educated about the effects of the changes.
All education should be documented and maintained properly.

6.6.1. System education

System operators require specific education in *UNICORN*, the computer's operating system (Windows NT), and the chromatography part of the system. Relevant and updated knowledge are essential for the proper functioning of *UNICORN* and the entire system.

6.6.2. Method education

When the system is designed, a plan for the education of the personnel should be established. The plan should specify the education requirements for each user function of the system. Education should be performed previous to the production start of the system. Verifying that all personnel have updated and adequate education for his/her function should be a part of the validation for the system.

6.7. Validation Approval

As mentioned in Section 1, General, the person responsible for the system should approve of its fitness for its intended use. This requires checking that everything included in the Validation Plan has been performed, documented and approved by the personnel responsible for each activity in accordance with the Activity Schedule.
Other things that are relevant to the functionality of the system, but not included in the Validation Plan, should also be included in the Validation Approval. This may include things have been discovered, or things that the persons involved in the validation process find relevant to the system.
All documents generated during the validation process and the findings of the persons involved in validating the system should provide information on the suitability of the system to start its intended production.

6.8. Validation of PC-based Networks

6.8.1. Introduction

The validation of a PC-based network is a complex and difficult task. In a regulated industry, one should consider this complexity prior to connecting a PC to a network. There is even a risk that the network benefits will not warrant the required validation efforts. There are three reasons for this:

1. The published literature only provides an overview of the principles involved in validation of networks. A great amount of work may be required to interpret and apply these principles to the individual network.

2. Regulators have not yet provided guidance on the requirements for computer networks, and it is, therefore, difficult to know exactly what is required for regulatory compliance.

3. There is a conflict between the "normal" use and handling of a PC network and its use in a regulated environment. In the latter case, the system must be validated and its use controlled by SOPs (standard operating procedures) that ensure every change is evaluated prior to its implementation. Situations may occur in which changes considered minor by network operators may be considered major by QA personnel. In a regulated environment, program version changes, upgrading of hardware, and additions of new functionality are all considered major changes.

   These three issues do not imply that it is impossible to validate a PC network and to maintain the desired quality level, but rather that the selected technical solution employ a minimum of hardware and software. The validation effort needed will most likely increase exponentially with the complexity of the network and the number of programs involved.

6.8.2. Validation strategy

The validation of a PC network can be performed by breaking it down into three units: applications (e.g., chromatography), network, and interfaces (where data is transferred from one unit to another). The same basic principles as those recommended for any computer system used in a regulated environment should be followed. But it is extremely important that the interfaces to other validated units on the network are specified and documented, and any other aspects that affect another unit must also be considered. After completion of the validation of these three units, the entire PC network is validated as a whole by testing it under "normal" situations as well as performing stress tests to verify the correct behavior of the system under different situations. Stress testing may include communication failure simulation and security challenges.

PC networks require an network operating system which handles the network communications, file management, user access, etc. The operating system has control over the network in much the same manner as an operating system in a PC, but is far more complex and accordingly, harder to validate. The same difficulties that may occur in the validation of a PC operation system can also be found in the validation of the network operating system, significantly complicating the validation effort.

The validation effort can be simplified by following the recommendations of PDA's committee on vendor evaluation*, namely, one can evaluate the network operating system vendor to assess their technical competence and ability to supply and support the proposed system. In addition, for widely used programs (and versions), one should perform an evaluation of the previous uses. For many applications this should indicate that the program is well functioning and reliable (e.g. MS-DOS or UNIX).

Some suggested validation documents and activities are presented in Table 1. Table 2. lists some SOPs that should be implemented for a validated network system.

Table 1 : Suggested Validation Documents/Activities

- Network description of all hardware and software. Printout and copies of all system-unique files, such as startup files and settings for the system and login scripts for users.
- Drawings, protocol used, drivers, netware, boards.
- Function tests, such as : communication, checksum tests (or other) of transferred data, stability and stress tests.
- Security, definition, tests.
- Test of backup-restore, RAID (or other system) SOPs.
- Evaluation of netware supplier*.
- Evaluation of hardware suppliers*.
- Network failures - consequences.
- Storage and access of documentation.
- Interfaces to other validated systems, tests and documentation.
- Education of users and network operators.

- Manuals - hardware, software, and system specific.
- Other relevant activities/documents to verify the function of the network.

*see also Section 7, page 84, ”Validation of Computer-Related Systems” section 4.3 and appendix 2.

Table 2 : Suggested SOPs

- Backup/restore - handling of backup media, frequency, storage, responsibility, generation handling of backups.
- How to add new users / deletes old, ascertain correct access levels for users.
- Authorization to add new users/change access levels/deletion of users. Documentation of access level changes for users.
- Security - routines and responsibility.
- New hardware (adding new PCs/printers and modifications of the net).
- Access times for the network.
- Responsibilities for the system.
- Changes in documentation.
- Start and stop of the system.
- ”Catastrophy routine”, including who authorizes catastrophy recovery activities.
- Frequent controls - volume controls, correct access levels for users.
- New soft and hardware - tests and how to document, approval criteria.
- Log book for the system - what to denote and how.
- Virus controls, frequency and anti-virus program to use.
- Revalidation procedure, frequency, data to use in a revalidation, expected results.
- Education of users.
- Manuals - updating, distribution, versions.

6.9. GMP references

This part contains references for items mainly in Section 6.
The applicability for each paragraph is somewhat unique for each system, and additions and/or modifications may be necessary. See also Introduction in this section.
The GMP can be found in USP XXIII, pages 1907 - 1922.

Item	GMP paragraph
1	§210.1
2	§211.63, §211.65, § 211.68, §211
3	§211.68
4	§211.68
5	All parts of GMP where applicable
5.5	§211.68(b)
5.7	§211.68(a)
6	§211.22- §211.28, especially §211.25
7	All parts of GMP where applicable

6.10. Installation Qualification and Operational Qualification Services from Amersham Biosciences Fast Trak Validation

In order to facilitate the successful start-up of systems at customer sites, Amersham Biosciences Fast Trak Validation has developed a comprehensive set of system qualification protocols. The protocols address process equipment and control system documentation according to the concepts developed by PMA. The qualification protocols are not a part of this validation support documentation. They are available separately from Amersham Biosciences Fast Trak Validation. A brief description of the various protocols is provided below:

6.10.1. Process Equipment

These protocols address the fluid handling parts of the various systems. Separate IQ and OQ protocols are available for *BioProcess* and *BioPilot* systems.

6.10.1.1. *BioProcess* Systems

As *BioProcess* systems are essentially custom designed and built to satisfy the unique requirements of an individual customer, the IQ and OQ protocols are based upon fully featured systems. For the majority of delivered systems, the user can directly apply the protocols, skipping those parts of the qualification which address features not present on their particular system. For highly customized systems, the protocols can be easily modified by the user or Amersham Biosciences Fast Trak Validation.
The protocols address: materials of construction, configuration audit, calibration, system cleaning and sanitation, maintenance, and existence of procedures.

6.10.1.2. *BioPilot* Systems

The relative standardization of *BioPilot* systems allows the majority of system purchasers' to utilize the qualification protocols without change. Where custom systems are provided, the situation parallels that described above for *BioProcess* systems. Since custom systems are usually only minimally different from the standard system, protocol adaptation is straightforward.

The IQ and OQ protocols for *BioPilot* systems address items similar to those described above for *BioProcess* systems.

6.10.1.3. *ÄKTA* Systems

Second part of the OQ is performed according to the Service Manual and supervised by a Service Manager from Amersham Biosciences. After the procedure a certificate is left with the customer.

6.10.2. Control System

The installation qualification of the control system components utilized in *UNICORN* is divided into two sections; one for the personal computer(PC) and a second addressing the components directly connected to the fluid handling unit. The operational qualification activities are addressed in a combined document for the PC and process unit. The OQ for *BioProcess*

and *BioPilot* systems are discrete documents reflecting the differences in the two systems.

6.10.2.1. *BioProcess* Systems

The IQ protocol utilizes an identical approach to that utilized for the fluid handling components. The model system utilized for the protocols is the same as that utilized in the fluid handling section, and the user options in protocol adaptation are identical.

The items addressed in the IQ protocol include: instrument identification, calibration, component identification, etc. The OQ document is described in the section on *UNICORN* .

BioPilot Systems

The IQ protocol is based upon a standard *BioPilot* system and can be easily adapted to accommodate the minor variations in *BioPilot* systems.

The IQ protocol for *BioPilot* systems address items similar to those described above for *BioProcess* systems. The OQ document is described in the section on the *UNICORN* which follows.

6.10.2.2. *UNICORN*

The IQ protocol for *UNICORN* addresses the personal computer utilized as the primary user interface. The protocol addresses system components such as hard drives, monitor, printers, memory, other installed software, etc. As the PC utilized for *UNICORN* is standardized with regard to minimum requirements, a single IQ document is utilized for both *BioProcess* and *BioPilot*. The OQ protocol focuses on the use of the *UNICORN* software to control the fluid components. The protocol addresses calibration, data accuracy, defaults, alarms, etc. The last portions of the OQ protocol use standardized methods to evaluate system performance.

7. UPGRADING OF *UNICORN*

7.1. General

For software which may affect product quality, it should be verified that it has been properly developed and is supported to meet the demands specified by the pharmaceutical producer, who is responsible for the program when it is being used as a component of pharmaceutical production.

In a regulated environment, upgrading of software should be carefully evaluated and the rationale for the upgrade documented prior to the implementation. The risk is that the cost and time needed for performing an upgrade, documenting, and testing it, may be higher that the benefits obtained with the new version.

7.2. Benefits vs. Costs

Before the decision is made to upgrade software, it should be determined that there is a need for the upgrade, and that the new version has obvious advantages which will give functional or other benefits, which make an upgrade relevant.

When a pharmaceutical producer upgrades from an older to a newer version of a software, the following costs can be assumed:

- Costs to assemble documents / procedures (time)
- Costs for preparations before the update (tests)
- Cost for the update itself (software and other components needed)
- Costs to test and verify the function of the system
- Lost production time.

These costs should be calculated and compared to the benefits of an updated program.

7.3. Revalidation / Change Control

Generally the activities needed to upgrade software should not trigger a total revalidation of the system. The intention of a complete revalidation is normally to verify that the whole system works as intended after a certain period of time. In this case, in addition to the updated software, other items are qualified.

In the case of upgrading the software, this could be seen more as something that should be subject to a change control procedure to ascertain that the system is capable of producing product with the same quality. Accordingly a specific change control procedure which is in line with the company SOPs and validation philosophy is preferable.

However, if a revalidation procedure exists, this should include tests to be performed to verify the correct function of the system. Tests from this revalidation procedure which are relevant for the verification of the system’s

function can be selected and used to qualify the system after the upgrade has been performed.
Examples of such tests which may be relevant to use from a revalidation procedure to verify the function of a software upgrade are:

- Test data sets for different program modules and expected results
- Evaluation of the backup / restore procedure
- Verification of instrument functions
- Verification of valve controlling functions
- Verification of the correct function of the methods used in the system
- Verification of alarms and warnings
- Verification of the signals to and from the instruments.

Examples of verifications which normally should be included in a revalidation procedure, but which are of little or no relevance in the case of a software upgrade are:

- Verification that the log-book has been used correctly
- Verification that security procedures has been followed
- Verification that calibrations has been performed according to SOPs
- Verification that sanitizations has been performed according to SOPs
- Verification that errors and disturbances has been reported.
- Verification that the personnel has received GMP and other education
- Verification that backups has been taken according to SOPs
- Verification that the SOPs are updated.

8. REFERENCES AND RELATED LITERATURE

:

8.1.1.1. References

1. ”Validation concepts for computer systems used in the manufacture of drug products.”
PMA’s Computer System Validation Committee, Pharm. Technol., (1986), Vol.10. No.5, 24-35

2. Good Manufacturing Practice, USP 23, pages 1907 - 1922.

3. ”Guideline for Process Validation”, May 1987, FDA, Rockville , MD, USA

4.”Computer system validation - staying current : Vendor-User relationships”, PMA’s Computer System Validation Committee, Pharm. Technol. (1993), Vol 5, No.8.

5. EN ISO9001 ed.1, 1994, SIS, Stockholm, Sweden

6. ”Validation of Computer-Related Systems”, PDA Committee on Validation of Computer-Related Systems, PDA Journal of Pharmaceutical Science & Technology, Vol.49, No.1/January-February 1995, Supplement

8.1.1.2. Related Literature :

”Computer System Validation - Staying current : Installation Qualification”, PMA’s Computer System Validation Committee, Pharm. Technol., (Sept. 1990), 88-104

”Computer System Validation - Auditing Computer Systems for Quality”, PhRMA (CSVC), BioPharm, Sept 1994, 22 - 31

IEEE Standard for Software Verification and Validation Plans”,
American National Standard, Institute of Electrical and Electronics Engineers, New York, NY, USA

”Real-Life Experiences in Computer System Validation”, Joseph M. George, Pharmaceutical Technology, Nov. 1994. 38 - 50.

”Good Automated Laboratory Practices”
United States Environmental Protection Agency, Office of Administration and Resources Management, NC, USA, (Dec. 1990)

”Good Automated Manufacturing Practice, Supplier Guide for Validation of Automated Systems in Pharmaceutical Manufacture”, Second Draft, UK Pharmaceutical Industry Computer Systems Validation Forum, (Jan. 95)

9. INDEX

P

Q

R

S

T

U

V

A