

## CHAPTER 7

# *Writing the User Requirements Specification*

The user requirements specification (URS) is the key document in the whole of the system development life cycle that is required for both business (investment protection) and regulatory reasons (defining intended purpose). Spend sufficient time defining and writing testable requirements.

## **7.1 What do the Regulators Want?**

### **7.1.1 FDA GMP and GLP Predicate Rules**

Both the GLP (§58.61)<sup>14</sup> and GMP (§211.63)<sup>12</sup> regulations require that equipment be fit for intended purpose; therefore, to define intended purpose a URS is required.

### **7.1.2 European Union GMP**

Annex 11, Clause 2<sup>27</sup> states:

*The extent of validation necessary will depend on a number of factors including the use to which the system is to be put, whether the validation is to be prospective or retrospective and whether or not novel elements are incorporated. Validation should be considered as part of the complete life cycle of a computer system. This cycle includes the stages of planning, specification, programming, testing, commissioning, documentation, operation, monitoring and modifying.*

### **7.1.3 FDA Draft Part 11 Validation Guidance**

This withdrawn FDA draft guidance<sup>23</sup> discusses the main points in the life cycle of a computerised system and makes the point that a system requirements specification (SRS) is required. In the FDA's view:

*Without first establishing end user needs and intended uses, we believe it is virtually impossible to confirm that the system can consistently meet them.*

Put in its bluntest form: without a requirements specification you cannot validate your CDS.

*Once you have established the end user's needs and intended uses, you should obtain evidence that the computer system implements those needs correctly and that they are traceable to system design requirements and specifications. It is important that your end user requirements specifications take into account:*

- *Predicate rules,*
- *Part 11, and*
- *Other needs unique to your system that relate to ensuring record authenticity, integrity, signer non-repudiation, and, when appropriate, confidentiality.*

Just in case, you think that a CDS is just commercial system and you can get away with doing little or nothing, you are wrong:

*Commercial software used in electronic recordkeeping systems subject to part 11 needs to be validated, just as programs written by end users need to be validated. We do not consider commercial marketing alone to be sufficient proof of a program's performance suitability.*

*The end user is responsible for a program's suitability as used in the regulatory environment.*

*However, the end user's validation approach for off-the-shelf software is somewhat different from what the developer does because the source code and development documentation are not usually available to the end user.*

## 7.1.4 PIC/S Guide

Section 9.2<sup>31</sup>:

*When properly documented, the URS should be complete, realistic, definitive and testable. Establishment and agreement to the requirements for the software is of paramount importance. Requirements also need to define non-software (e.g. SOPs) and hardware.*

## 7.1.5 General Principles of Software Validation

Section 5.2.2<sup>29</sup>:

*The software requirements specification document should contain a written definition of the software functions. It is not possible to validate software without predetermined and documented software requirements.*

## 7.1.6 Regulatory Summary

A URS is essential for the validation of any CDS operating in a regulated environment. Requirements must be testable, traceable and some requirements may indicate that a procedure needs to be written.

## 7.2 Business Rationale for Writing a URS

How much money does your organisation waste on buying computer systems that do not work or do not meet their initial expectations? The number of CDS systems

that have no or inadequate user requirements typically outnumber the systems that have adequate specifications. A well-written URS provides several specific benefits, as it:

- Serves as a reference against which off-the-shelf commercial products are selected, evaluated in detail, and any enhancements are defined. You are less likely to be seduced by technology or buy a poor system using this approach.
- Reduces the total system effort and costs, since careful review of the document should reveal omissions, misunderstandings and/or inconsistencies in the specification and this means that they can be corrected easily before you purchase the system.
- Provides the input to user acceptance test specifications and/or qualification of the system.

A URS defines clearly and precisely, what the customer (*i.e.* you) wants the system to do, and should be understood by both the customer and the instrument vendor. The URS is a living document, and must be kept updated, *via* a change control procedure, throughout the computer system life cycle. After purchase, if and when you upgrade the software, the URS is also updated to reflect the changes and new functions in the latest version.

A URS defines the functions to be carried out, the data on which the system will operate, and the operating environment. Ideally, the emphasis is on the required functions and not the method of implementation as this focuses on the “what” rather than the “how”.

If you are selecting a new CDS, then the main purpose of a URS from a business rationale is to select a system based on your laboratory’s defined requirements. This avoids the user community from being seduced by technology as the system selection is based on documented requirements and what the users actually want. Furthermore, it allows the selection to be based on objective requirements that will allow you to cut through the marketing literature and focus on your specific requirements. As the URS defines what the system can do, it provides a platform to assess if any system can provide the required functionality.

## **7.3 Contents of a Chromatography Data System URS**

### **7.3.1 When to Write the URS**

The URS is usually the first document to be written in the life cycle validation of a CDS. The rationale for this approach is that some of the requirements specified might impact the validation strategy for the system such as a phased roll-out and this will need to be written into the validation plan.

### **7.3.2 Link the URS to a Specific Software Version**

The URS for a CDS, or indeed any computer system, is a living document and must be linked to the specific version of the application software that is being validated;

for example, ChromSystem version 4.1. If the system is being updated to a new version, the URS must be reviewed and revised to be applicable to the new version of the software (*e.g.* ChromSystem version 5.2). Therefore, ensure that the version number of the CDS application software is written in the title and introduction of the document as a minimum.

### 7.3.3 Sections of the URS

From the V model shown in Figure 12 you can see that the user requirements are related to the tests carried out in the qualification phase (typically either the OQ or the PQ). Therefore, it is important to define the requirements for the basic functions of the CDS, the adequate size, 21 CFR 11 requirements and consistent intended performance in the URS that will be tested before the system goes live.

The main elements in a URS are:

- Corporate requirements for hardware, workstations and operating systems, *e.g.* terminal emulation such as Citrix.
- Overall system requirements such as number of users, locations where the system will be used and the instruments connected to the system.
- Compliance requirements from the predicate rule and 21 CFR 11 such as:
  - Open or closed system definition
  - Security and access configuration of the software application including user types
  - Data integrity
  - Time and date stamp requirements
  - Electronic signature requirements
- Defined data system functions, which should be based on the CDS workflow outlined as in Figure 5 is the best framework for writing a URS. Therefore, if you have mapped the process (Chapter 6), this makes an ideal reference and prompt for formulating the requirements as they can be defined against each activity in the process. In addition, include requirements for system capacity such as maximum number of samples to be run, custom calculations and reports for the initial implementation and roll-out, *etc.*
- IT Support requirements. These include backup and recovery, off-line archive and restore.
- Interface requirements. For example, will the CDS be a standalone system or will it interface with a LIMS and if so how?

Table 4 presents a suggested list of the main sections of a URS for a CDS. This is a robust approach based on a number of validations of systems in many organisations. For a large client-server or terminal server CDS system there can be up to 500–600 requirements depending on the nature of the work that the system automated.

This idea of documenting what we want in sufficient detail sounds great, but it means more work, does not it? Yes, this is true but consider the overall benefits to you and the laboratory. The more time you spend in the specification and design phase getting your ideas and concepts right the quicker the rest of the life cycle will

**Table 4** *Suggested contents of a user requirements specification for a CDS*

<i>Section</i>	<i>Contents</i>
Introduction	Purpose and scope of the document Referenced documents
IT requirements	Hardware specification for workstations and servers Operating system and database specification, <i>e.g.</i> Oracle 9i and version number
System requirements	Outline system capacity defined, <i>e.g.</i> number of users, chromatographs to be interfaced
Compliance needs	Open or closed system definition for the CDS Predicate rule and 21 CFR 11 requirements for e-records Predicate rule and 21 CFR 11 requirements for e-signatures if used
CDS functions	Operation of the system in the laboratory from set-up, instrument control, data acquisition, integration and review of the data, SST calculations and acceptance criteria, calibration models used, results calculations and reporting
Support needs	Backup and recovery of the system Housekeeping User account management
Interfaces	Is the CDS standalone or interfaced with a LIMS: if the latter then what, how and when data are transferred between the two systems
Data migration	Migration requirements from existing version to new version of same software Migration from old to new application with old system retirement
Appendices	Glossary Terms and definitions

go, as you know what you want. You will get a CDS that meets your requirements more fully and rather than find out when the system goes live that it cannot perform certain functions.

### 7.3.4 General Guidance for Writing the Requirements

The following guidelines should be followed during the production of the specification:

- Each requirement statement should be uniquely referenced and no longer than 250 words.
- The URS should be consistent and requirement statements should not be duplicated or contradicted.

- Specify requirements and not design solutions. The focus should be on what is required, but not how it is to be achieved.
- Each requirement should be testable. This allows the tests to be designed as soon as the URS is finalised.
- Both the customer and the vendor must understand the document. Therefore, jargon should be avoided wherever possible and key words are defined in a specific section in the document.
- Requirements should be prioritised as mandatory or desirable.
- The URS should be modifiable but changes should be under a formal control procedure.

A URS is correct if every stated requirement has only one interpretation and is met by the system. Unfortunately, this is very rare.

### 7.3.5 URS Issues to Consider

When defining your requirements some or all of the following will need to be included in the URS depending on your ways of working and the chromatography instrumentation that you will connect to the CDS:

- Data capture rates for all chromatographic techniques connected to the CDS will need to be specified. For example, conventional chromatography with a run time in the order of 20 min requires a data capture rate of 1 Hz. However, for capillary GC 10–20 Hz is more appropriate and for CE a higher rate still may be required depending on the overall migration time, analyte peak shape and width.
- Specify the calibration models that you will or intended to use. There are many different types of calibration model available in a CDS, however you only need to validate those models you actually propose to use; these need to be documented in the URS and all other calibration models are excluded from the validation. There is a downside in this approach, if a new calibration method is used that has not been validated then there is a regulatory risk and any new calibration method needs to be validated after an approved change control request. However, this can be managed *via* the change control process that is described in Chapter 20.
- Depending on your data system, several chromatographs may be linked into a collection workstation or an A/D unit. Here, consider if crosstalk (the interference from one channel to another) could be an issue if the A/D chip is multiplexed across two or more channels. Alternatively, consider if the total sampling capacity of the data collection and buffering unit or server is adequate for the proposed systems.
- Has the maximum number of injections for an analytical run been defined? This is a critical component, if 100 vials are routinely injected in a run, the system cannot be tested with a run of only ten samples as a user has not demonstrated adequate size or adequate capacity. The specification must match the use of the system including replicate injections.

- Some data systems will be configured to collect data from diode array detectors (DAD). If this is required, especially to analyse product, then the data collection and analysis will need to be checked as part of the adequate size as some data files can be in the Mb range. The file delete option should not be enabled to protect the electronic records generated.
- Virtually all client–server CDS systems will have a buffering capacity within their A/D or data collection units. Therefore, part of the adequate size requirements must be the ability to capture and buffer data if the network is unavailable, followed by the successful transfer of data to the server when the network connection is re-established.
- How many users will there be on the system at the same time and will the system still perform its functions reliably? This number may be lower than the number of concurrent users that you have a licence for but this is a major requirement to define in the URS and test during the PQ. If the system becomes unreliable or unstable as the number of users increases then the system owner cannot state that the system has adequate size or can perform as intended.

These are some of the considerations for each installation of a CDS. Once installed in a laboratory environment and on the organisation's network it becomes unique. The number of users, network components, server components, operating systems, software patches and laboratory configuration make it so, therefore you need to demonstrate that it works under your operating environment.

### 7.3.6 Making the Requirements Traceable

Although not mentioned specifically in the regulations but stressed in the various guidance documents,<sup>1</sup> traceability of system requirements to the testing phase is important for any system including a CDS. Therefore, the way that system requirements are presented and managed is important. It is all very well the regulations stating that a user must define their requirements in a URS, what does this mean in practice? Table 5 illustrates one way that capacity requirements can be documented; each requirement. Note that each requirement is:

- Uniquely numbered
- Written so that it can be tested, if required, in the PQ.
- Prioritised as either mandatory (M = essential for system performance) or desirable (D = nice to have and the system could be used without it). This prioritisation can be used in risk analysis of the functions and also for tracing the requirements through the rest of the life cycle as will be discussed in Chapter 12.

Remember that the URS functions are related to the tests carried out in the qualification phase of the life cycle. Therefore, if you have not specified the requirements in this document; how can you test them?

**Table 5** *How system requirements for CDS capacity can be documented*

<i>Req. No.</i>	<i>Data system feature specification</i>	<i>Priority M/D</i>
3.3.01	The CDS has the capacity to support 10 concurrent users from an expected user base of 40 users	M
3.3.02	The CDS has the capacity to support concurrently 10 data A/D data acquisition channels from an expected 25 total number of channels	M
3.3.03	The CDS has the capacity to support concurrently 10 digital data acquisition channels from an expected 25 total number of channels	D
3.3.04	The CDS has the capacity to control concurrently 10 instruments from an expected 20 total number of connected instruments	M
3.3.05	The CDS has the capacity to simultaneously support all concurrent users, data acquisition and instrument connects whilst performing all operations such as data reprocessing and reporting without loss of performance (maximum response time is < 10 s from sending the request) under peak load conditions	M
3.3.06	The CDS has the capacity to hold 70 GB of live data on the system	D

The key point for traceability is that requirements are individually numbered. Requirements traceability will be discussed further in Chapters 12, 14, and 18.

### 7.3.7 Reviewing the URS

Ideally, an independent group of users (persons not involved in writing the document) should evaluate the URS and challenge each requirement including any interfacing requirements for chromatographs or any other computer applications. If any missing requirements or inconsistencies can be found at this stage they are usually easy and inexpensive to correct. Therefore, the extra work in ensuring that the URS is correct are time and resources well spent. Problems that can be rectified at this stage are far cheaper to solve than those identified later in the life cycle. When the URS is complete, the outline selection tests can be generated that can be used to select a potential system and reused later in the life cycle during the PQ testing.

## 7.4 Writing Testable Requirements

The key to a well-written URS for a CDS, or any other computerised system, is testable requirements. However, a major problem is that many CDS users do not have an idea of how to write a user requirement and hence the quality of the overall validation effort falls at the first fence. In this section, we will look at the ways to write testable requirements.



### 7.4.1 How not to do it

To illustrate some of the problems of writing testable user requirements, here are two examples of how not to write requirements for a CDS. Note that both requirements are uniquely numbered which is good.

Performance Issue 6.1.8.1: Operating at *normal* PC response times with no *undue* delay in response at *low* computer utilisation.

In requirement 6.1.8.1 there is the use of wording that makes the requirement untestable. The words normal, undue and low render the requirement useless and incapable of being tested (*e.g.* weasel words).

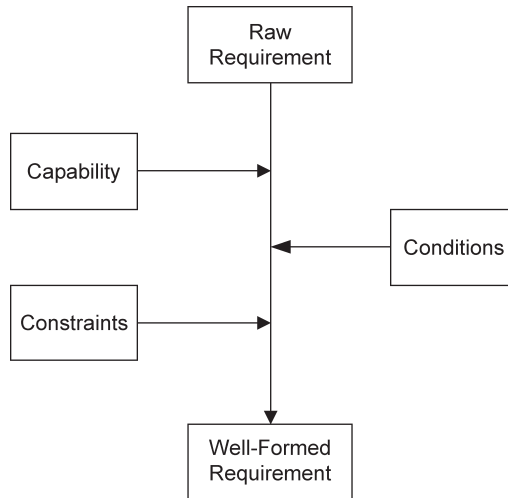
Reporting 6.2.4.1: Report production at a rate of at least a page every 10 s at *modest* network and server utilisation.

Requirement 6.2.4.1 is marginally better as “report production at a rate of at least one page every 10 s” is testable and specific. However, the requirement then snatches defeat from the jaws of victory with the phrase “at modest network speed” which renders it untestable as “modest” cannot be defined. Therefore, we need to have a better approach to writing user requirements.

### 7.4.2 Writing Well-Formed and Testable Requirements

The recommended guide for writing software requirements is IEEE Standard 1233, entitled “A Guide to Writing Software Requirements”,<sup>50</sup> which states that a well-defined requirement must have the following attributes:

- *Capability*. Capabilities are the fundamental requirements of the system and represent the features or functions of the system needed or desired by the user. A capability should usually be stated in such a way that it describes what the system should do and in a way that is solution independent.
- *Condition*. Conditions are measurable quantitative attributes and characteristics that are stipulated for any capability. A condition further qualifies what is required from a capability and allows the capability to be designed, evaluated or tested. This is the critical element that is usually missing from a requirement.
- *Constraint*. Constraints are requirements that are imposed on the solution by circumstance, force or compulsion, *e.g.* regulatory or corporate standards. Constraints absolutely limit the options open to the laboratory of a solution by imposing non-negotiable boundaries and limits. Examples of constraints can include interfaces to already existing systems where an interface cannot be changed, the need to change passwords regularly and the use of a specific operating system or database to meet corporate standards. Constraints can be written either as standalone requirements themselves (*e.g.* the database will be Oracle) or as constraints upon individual capabilities (as shown in the password example below).



**Figure 27** Writing well-formed requirements (taken from IEEE Standard 1233)

To illustrate the process of how to write well-formed requirements, see Figure 27.

Therefore, an example of a well-formed requirement for a CDS is the need for a password that can be used to access the system as well as be part of the component of an electronic signature. The three elements needed are shown below:

- *Capability*: the system will have alphanumeric passwords
- *Condition*: the length of which will be a minimum of six characters
- *Constraint*: and these will be changed automatically every 90 days (to meet regulatory requirements and corporate IT policy)

### 7.4.3 Key Criteria for User Requirements

A well-formed requirement<sup>50</sup> is a statement of system functionality that:

- Can be tested
- Must be met or possessed by a system to solve a user problem or to achieve a user objective
- Qualified by measurable conditions
- Bounded by constraints

Therefore, armed with this knowledge and a little practice your ability to write better requirements should improve.

## 7.5 Documenting System Configuration and Customisation

If you are purchasing a new system there will be not be the information available to include in the URS some of the configuration aspects of the CDS. However, if

the system is already installed and you are either validating an upgrade or retrospectively validating your existing system, then you have an option to include some of the system configuration details in the URS.

Note the use of the word option. It can be included in the URS where you can add sections on the access control and user types needed along with the access privileges. In addition, the URS can contain the system configuration details, *e.g.* turning electronic signature functionality on or off as well as some custom calculations if required.

An alternative approach can be to have this information in separate configuration documents (some people may refer to these as either functional specifications or design specifications). There is no right or wrong way to do this as long as it is documented.

This part of the specification of the system must be kept up to date. If you are validating a new system these requirements may change as you gain understanding about how the system operates. However, there is a chance that if fixed early in the life cycle, they may become outdated by the time the system is rolled out.



<http://www.springer.com/978-0-85404-969-1>

Validation of Chromatography Data Systems

McDowall, R.D.

2005, 260p., Hardcover

ISBN: 978-0-85404-969-1

A product of Royal Society of Chemistry