

Pharma IT journal

The dedicated publication for those working with **Computerised Systems, Processes and Software** in the **Pharmaceutical, Biotechnology, Medical Device, Clinical Research and Supporting Industries**

Vol. 1 · No. 3 · July 2007

www.PharmaIT.co.uk

The Software Patching Dilemma

Qualification of Windows Servers (Part 2)

Best Practice Guide for Compliance Management

Specification of Excel Spreadsheets

Regular Features



Letter from the Editor



Laboratory IT Column



Compliance Corner

A Pragmatic Approach to the Specification of Excel Spreadsheets

GxP critical spreadsheets need to undergo specification to ensure that the user's needs and the spreadsheet's functionality is clearly defined and documented. This paper describes a pragmatic approach to the specification of Excel spreadsheets using a single generic document. The approach described is easily transferred to other simple systems such as databases and standalone instrumentation.

By David Harrison & David A Howard

Key Words: Validation, Compliance, Spreadsheets, MS Excel, 21 CFR Part 11, Pharmaceutical, GAMP, GxP, GLP, GMP, GCP, End User Computing, Specification.



Introduction

This continues the series of short articles describing a generic process for validating Excel Spreadsheets. A previous article¹ gave an overview to the process, whilst a subsequent topic will cover the testing and qualification.

Although this article specifically covers Excel, the principles can be adapted to cover a wide variety of applications. This process has been successfully used on Access Databases, and other straightforward applications such as standalone instrumentation and off the shelf software packages⁸. The focus is on minimising the document set, and providing all relevant information in a single generic specification with all critical items covered.

This paper outlines the approach, and then provides a set of questions and answers on the approach. The answers attempt to pre-empt any regulatory or QA questions that may arise.

The Streamlined Specification Approach

The recommended approach is summarised below.

- One single specification is used which incorporates the User Requirements Specification (URS), Function Specification (FS) and Design Specification (DS).

- The specification identifies each requirement with unique numbering so that they can be later used for traceability and qualification cross referencing.
- Where possible, information is collated into tables and appendices to ensure a consistent and easy to understand format.
- The process acknowledges that most spreadsheet specifications are generated retrospectively from an existing prototype. As such, the detailed content of the prototype is used in the specification process to allow the reader of the specification to better understand the spreadsheet and its functionality.
- The version of the spreadsheet is clearly defined in the documentation.
- A generic template document is used as the starting point which is suitable for all spreadsheets with minimal modification.

Specification Generation Process

The specification document is generated retrospectively once the **final version of the prototyped spreadsheet is complete**. This is critical, as one common error is to generate the content, only to see the users change their mind on functionality, resulting in changes to the prototype. This 'scope creep' results in time consuming changes to the documents and an increased likelihood of errors.

Specifications are usually generated by the spreadsheet developer and reviewed independently by a QA or validation function.

Once generated and approved the specification (coupled with the spreadsheet itself) is presented for qualification.

The specification will act as a living document, and future changes to the spreadsheet (through change control) will result in version controlled updates to the specification.

Specification Content

The specification document is routinely divided into two distinct sections, a URS Section and a FS section. The approval signatures on the front page approve all content.

ABB managing compliance enabling performance

ABB is an international technical consultancy providing a range of services including licence to operate, asset management, manufacturing improvement and capital investment to customers in the Life Sciences business sector.



ERP Systems

ABB is a world leader in helping clients manage and implement ERP systems to achieve early success and payback from IT investments. Our proven SAP validation approach centres on a holistic analysis of how the system is integrated with the business operation, directly aligning business and project aims. Combining experienced leadership and our ERP project accelerators we can deliver high quality ERP validation projects at up to 30% less cost.



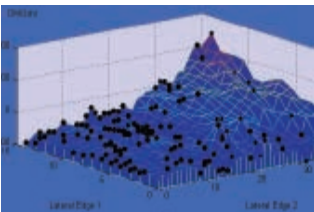
End User Computing

End User Computing provides the optimal environment for users, however the inherent flexibility can bring compliance concerns and lead to extended validation effort. ABB are the world leaders in spreadsheet and database compliance, providing cost-effective and pragmatic solutions through the use of technical controls and efficient validation processes. Our proven turnkey solutions ensure your systems meet both regulatory and user requirements.



Computer Systems

ABB has a proven track record in providing world class consultancy and validation on Computer Systems (DCS, PLC, WMS, LABS, LIMS etc) throughout the pharma industry. Whether on new implementations or upgrades, the combination of our widespread experience and risk based approach to validation ensures a cost effective, pragmatic solution that fits your business, resource and budget needs.



PAT

Quality by design (QbD) concepts based on Process Analytical Technology (PAT) is fast becoming a corporate development imperative for many of our pharmaceutical clients. ABB are at the leading edge of this drive, combining our excellent technology, regulatory consultancy and change program management skills to offer the perfect technology partnership for our clients.

The document is divided up into the following sections.

Section 1 – Introduction

Section 1 provides an introduction to the project and the goal of the project.

Section 2 – System Overview

- 2.1 User Background
- 2.2 System Overview
- 2.3 Specification Methodology

Section 2 provides information on the spreadsheet's use such as departmental background and how the spreadsheet fits into the data processing/approval process. It will define the data used and records generated. It will provide an overview of the spreadsheets use and function.

This section also provides a brief introduction to the format and content of the document with particular note made to the use of Appendices.

Section 3 - User Requirements

Section 3 details the user's needs and requirements. Although it is generated retrospectively, it is still written as a request for functionality i.e. The system **must** do this, or the system **should** do that.

Section 4 – Functional Specification

Section 4 details the response to the User Requirements. It is generated retrospectively and provides a detailed description of the functionality defined within the spreadsheet. It is written in the present tense i.e. The system **does** do this.

In sections 3 and 4 each item is structured into numbered bullet points which allow cross reference and traceability. The majority of bullets are requested in the URS section, and answered in the FS section.

e.g.

URS Section 3.1.2 The spreadsheet must be compatible with Microsoft Excel 2002 and 2003.

FS Section 4.1.2 The spreadsheet/template can be used within the following Microsoft Excel application software versions (Excel 97, Excel 2000, Excel XP, Excel 2003).

For many of the requirements this functional response is a straightforward confirmation of the need; for other requirements the FS section of the document will expand the detail to explain how the functionality works.

The following descriptions detail the relevant subsections of sections 3 and 4; the descriptions are combined in this paper, but are separate in the specification document.

URS 3.1 and FS 4.1 – Application Software

Describes the versions of Excel (and any other relevant software).

URS 3.2 and FS 4.2 – Spreadsheet Workbook Characteristics

Describes the way in which the spreadsheet is structured into workbooks and worksheets. It will define the number of each and whether they operate as XLS or XLT files.

URS 3.3 and FS 4.3 – Location of Operation

Describes the proposed location of operation such as

identification of the server and whether the spreadsheet will be operating within a 3rd party software package for added security².

URS 3.4 and FS 4.4 – Electronic Record Reproduction

Describes the needs and functionality over the use and availability of electronic data and records to meet 21 CFR Part 11³.

URS 3.5 and FS 4.5 – Electronic Record Backup and Restore

Describes the needs and functionality for accessibility of saved and restored electronic records to meet 21 CFR Part 11³.

URS 3.6 and FS 4.6 – Audit Trail

Describes the needs and functionality for audit trails and traceability to meet 21 CFR Part 11³.

URS 3.7 and FS 4.7 – Physical and Logical Security

Describes the needs and functionality for security to meet 21 CFR Part 11³. This will describe the security around both the application software (Excel and possibly a 3rd party software package²) and the individual spreadsheet. It will provide detail on workbook, worksheet and reference to cell protection.

URS 3.8 and FS 4.8 – Macro Functionality

Describes the needs and functionality of any macros used in the spreadsheet. Depending upon the complexity of the macros used, this section can follow a simple or complex structure. The complex structure will break out functionality into individually numbered sections which details Functionality type (i.e. button, form, etc), VBA command, Use, associated error messages and cause, and audit trail entries.

URS 3.9 and FS 4.9 – Spreadsheet Documentation

Describes the supplied documentation such as specifications, SOPs for routine use and the presence of developer and user training records.

URS 3.10 and FS 4.10 – GxP Records and Approval Signature Functionality

Identifies GxP critical electronic records to meet 21 CFR Part 11³ and describes the process for handwritten or electronic approval of the data.

URS 3.11 and FS 4.11 – Data Calculation Requirements

Describes the detailed functionality of the calculations and formulas within the spreadsheet. This section is usually substantial and always unique to each spreadsheet, as such the content is transferred out to suitable tables within Appendix A for the URS and Appendix B for the FS. See examples below.

URS 3.12 and FS 4.12 – Input/Output Requirements and Data Flow

Describes the detailed functionality of input and output such as Excel data validation, conditional formatting, and the process by which any data is imported/exported in the spreadsheet. For many of these functions cross reference can be made to Appendices A/B or to macros detailed elsewhere in the document.

URS 3.13 and FS 4.13 – Information Printout

Describes and displays the visual representation of the spreadsheet in use and any printed spreadsheet output. This section is usually covered by screen prints and printouts which are provided in Appendix C.

URS 3.14 and FS 4.14 – Miscellaneous Requirements

Describes the needs and functionality for any items not covered in previous sections. Example may include information on any manual processes that link this spreadsheets output to another system.

Section 5 – Glossary and Section 6 – References

Describes the acronyms and references used. Often the references will refer back to the testing of other applications such as any testing of the Excel environment or 3rd party software packages².

Appendix A – User Requirements Table Details

Describing how a spreadsheet works is difficult to put into words, especially if the spreadsheet is complex and has multiple interlinked tables and look ups. Experience has shown us that although wordy descriptions work fine for simple spreadsheets, they quickly become unmanageable and confusing for larger spreadsheets. This problem is compounded by the fact that two people will describe the functionality very differently in words.

Our approach to defining user needs and calculations is done using tables as shown below (Table. 1). Worksheets are broken down into tables and each table's need is defined by the user. Unique references for any calculations are defined at this stage to allow the start of the traceability through FS and qualification. Note that only **verifiable** 'calculation' cells are specified in this user requirements section; this would include formulae, conditional formats, data validation and graphical ranges. Data input cells and constant text cells are not included.

The user requirements are also supported at this stage by a fully functional prototype of the spreadsheet (printouts of which are included in Appendix C of the specification), and therefore this table, combined with the visual image allows the reader to get a clear picture of the user's needs and how the spreadsheet works and processes data.

This method may seem overcomplicated for such a simple example, but in reality spreadsheets are never this straight forward, and wordy descriptions are extremely cumbersome. From our experience this provides a consistent and efficient method of listing user requirements.

Appendix B – Functional Specification and Worksheet Contents

This section describes the spreadsheet content. Wordy descriptions do not work, and that the simplest way to cover this is to printout the whole of the spreadsheets content.

This can be done manually using a display option in Excel **(Tools, Options, View, Show formulas)** (Fig. 1).

Alternatively this output can be automatically generated by commercially available reporting tools such as Exchecker⁴, or Power Utilities Pak⁵.

Our approach is performed using a combination of these products and provides us with an output direct into MS word as shown below. This example table has been cut down to show example lines only.

Similar tables will exist for a variety of detailed content such as File Information, Links, Names, Comments, Data Validation, Conditional formatting, Charts, macros etc.

Appendix C – Workbook Display & Printouts

This section includes visual images of how the spreadsheet looks on screen and how any printed output looks. It is invaluable in allowing the reader to get a clear picture of the how the spreadsheet works and processes data. This section is also vital as the printed output is often the information that is submitted for regulatory approval and storage.

Appendix D – Macros Included In Workbook

This section includes the source code printout of any macros included in the spreadsheet.

Appendix E – Training Records

This section includes evidence of the training records of developers of the spreadsheet as requested by 21 CFR Part 11³. For simple spreadsheets this would include evidence of User Training in the use of Excel and in the requirements of 21 CFR Part 11 and computer systems validation.

For spreadsheets with macros it would include evidence of programmer qualifications and experience in VBA.

| Table A.1 | | Worksheet Name: Data Sheet | | Table Name: Header Information and Calculation | |
|---|-----------------------|----------------------------|---|--|--|
| This table contains details on the electronic file and spreadsheet calculation. It provides a mean and StDev of the data input values and provides a final results of mean value multiplied by concentration. | | | | | |
| Calc. Ref. | Cell Name/Description | No. Cells | Comments | | |
| A.1.1 | Date/Time Stamp | 1 | Must displays MODIFIED_TIMESTAMP field generated from the DaCS system. Timestamp must update every time the spreadsheet content is changed or updated. | | |
| A.1.2 | Filename Stamp | 1 | Must displays FILENAME field generated from the DaCS system. Filestamp must update every time the spreadsheet content is changed or updated. | | |
| A.1.3 | Assay Description | 1 | Data Validation Lookup - User must be able to select the assay type about to be reported upon from a drop down menu. Choices include. Assay 1, Assay 2, Assay 3. | | |
| A.1.4 | Mean Result | 10 | Table must provide a mean result for each row of data input values. | | |
| A.1.4 | StDev Result | 10 | Table must provide a standard deviation for each row of data input values. | | |
| A.1.4 | Final Result | 1 | Table must provide a calculation of the average of the (Mean Result * Concentration Factor) . | | |

Table. 1 User needs and calculations defined in tabular format

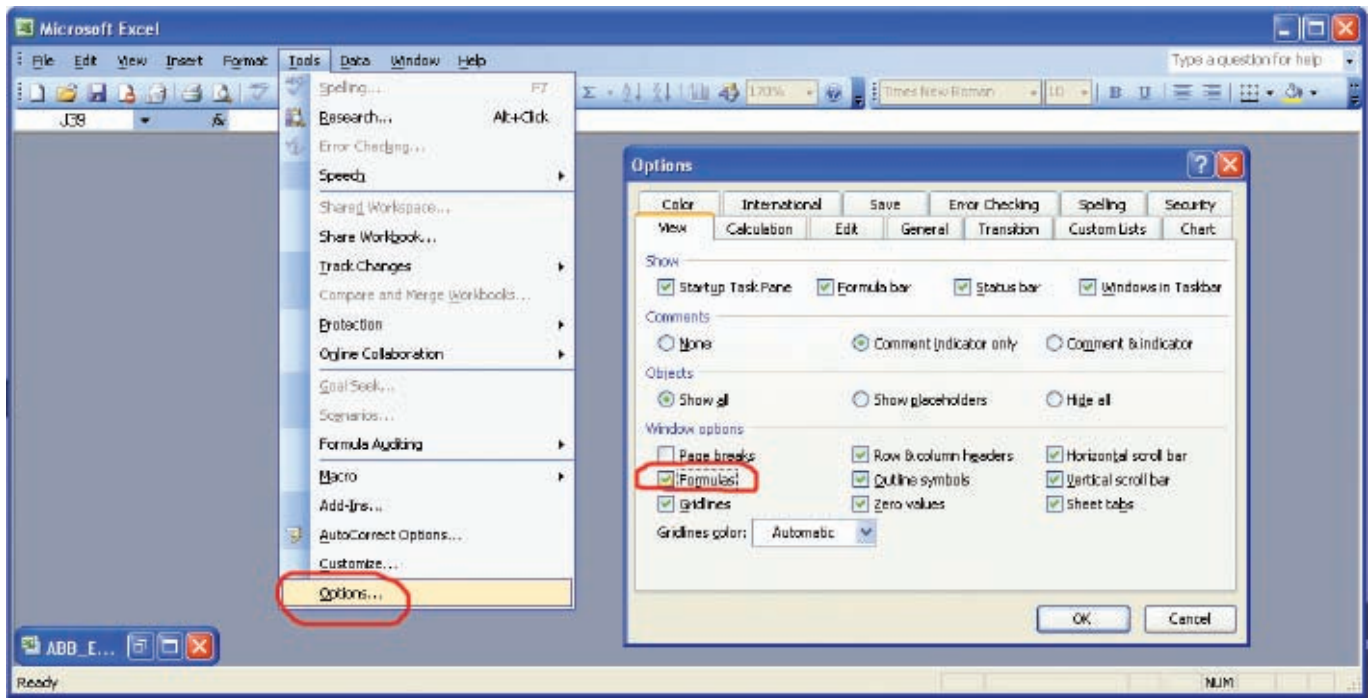


Figure. 1 Printing the spreadsheet content

| Cell Address | Cell Format | Cell Type | Formula / Value | Locked | Hidden | Merged | Merged Range |
|--------------|-------------------------|------------|--|--------|--------|--------|--------------|
| D2 | General | Constant | Spreadsheet Name: | True | False | No | |
| E2 | General | Constant | ABB Example | True | False | No | |
| B5 | General | Constant | Assay Description | True | False | Yes | B5:H5 |
| B6 | General | Data Entry | | False | False | Yes | B6:H6 |
| C9 | General | Constant | Weight of Item 1 | True | False | No | |
| F9 | General | Constant | Concentration Factor | True | False | No | |
| G9 | General | Constant | Mean | True | False | No | |
| H9 | General | Constant | SD | True | False | No | |
| C10 | 0.00 | Data Entry | | False | False | No | |
| F10 | 0.000 | Data Entry | | False | False | No | |
| G10 | 0.00 | Formula | =IF(ISERROR(AVERAGE(C10:E10)),"",AVERAGE(C10:E10)) | True | False | No | |
| H10 | 0.00 | Formula | =IF(ISERROR(STDEV(C10:E10)),"",STDEV(C10:E10)) | True | False | No | |
| B21 | General | Constant | Final Result | True | False | No | |
| C21 | 0.00 | Formula | =IF(ISERROR(AVERAGE(G10:G19)*AVERAGE(F10:F19)),"",AVERAGE(G10:G19)*AVERAGE(F10:F19)) | True | False | No | |
| B23 | General | Constant | Filename: | True | False | No | |
| C23 | General | Formula | =IF(ISBLANK(FILENAME),"",FILENAME) | True | False | Yes | C23:H23 |
| B24 | General | Constant | Date: | True | False | No | |
| C24 | dd mmm yyyy hh:mm:ss | Formula | =IF(ISBLANK(MODIFIED_TIMESTAMP),"",MODIFIED_TIMESTAMP) | True | False | Yes | C24:H24 |

Table. 2 Example of printout

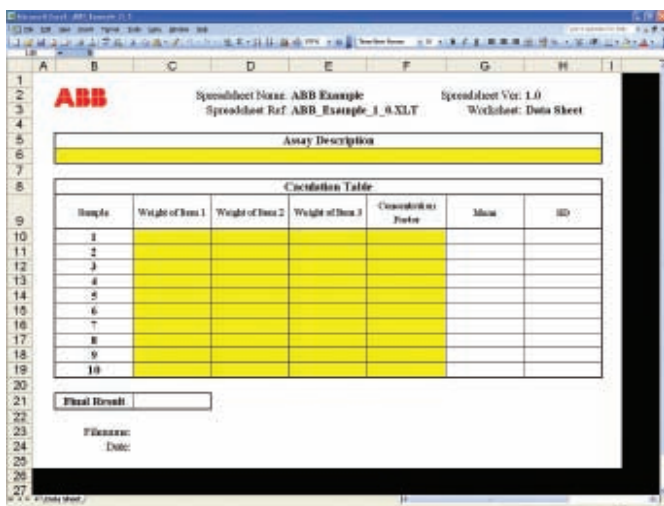


Table. 3 Workbook and display printout

Appendix F – Risk Assessment

This section documents the functional risk assessment (FRA) of the spreadsheet. The risk assessment will highlight risks associated with the use of the spreadsheet; each identified risk will be prioritised and critical risks will be mitigated. The standard approach used is GAMP⁶ based and very similar to the Risk Assessment Approaches for the Validation of Commercial Computerised Systems⁷ identified by other authors.

Challenging The Streamlined Approach

The approach recommended above is a simplified process compared to what you would normally undertake for specifying a major computer system, and as such normally generates a number of questions. The following questions and answers pre-empt those questions and give the authors view on why the approach is ideally suited for the applications we are dealing with.

1. Question. Is there any regulatory document that states you can / cannot combine the URS with the FS?

No, not that we are aware of! Companies will be expected to document User Requirements, and how the system functions. How you document it is not normally defined, only that it is clear, accurate and allows traceability.

2. Question. Our QA policy states that we must have a separate URS!

The recommended approach is an attempt to save you time and resources by streamlining the process. If your policies state that you must work in a certain way then unfortunately you will have to follow them. Being out of compliance with your own procedures is a serious breach, and one which is difficult to defend.

The simple answer to this dilemma is that you should modify your corporate policies and procedures to allow such a streamlined approach **where appropriate**. Most companies allow this approach, and it is key in the new "risk based" validation philosophies that people take.

Updating and improving your corporate policies in this way will benefit other projects in the future.

3. Question. Normally the User generates the URS and the software supplier generates the FS. How does this fit when we do it all ourselves?

The concept of user and supplier does not fit well for spreadsheets, as it is often the user who developed the spreadsheet. A division between user needs and developing never occurred as the development was undertaken as a prototyping activity.

Most spreadsheets are developed using a rapid application development (RAD), using prototypes and trial and error until it is ready to use. This approach is common and acceptable as long as you build your validation process around it.

When you prototype in this way the common practice is to retrospectively generate your URS and FS. Retrospectively generating the FS is simple and makes perfect sense, whilst retrospectively generating a URS often seems strange as you list what you have, rather than what you need.

Your auditor wants to see user needs and the actual operation specified, how you get to this situation is secondary.

4. Question. Can we generate separate specification documents and still follow the approach being recommended?

Of course. Returning to the traditional approach of having separate documents for URS and FS is common, but will take additional work, and therefore cost you more in money and resources. It will also slow down the approval and review process. Experience has showed that the document review and approval process is often a bottle neck in the validation process.

5. Question. The headings recommended don't match our corporate procedures on what should exist in a specification document.

The headings provided are the generic ones and often companies wish to change them to match up with their own corporate standards which is fine. These headings are provided to ensure you cover the key subjects and items that need to be specified.

6. Question. A generic specification implies that the content is the same for many of our spreadsheets. All our spreadsheets are very different.

Every spreadsheet is different, however many of the underlying requirements from a spreadsheet are the same or very similar.

For example,

- Your need to have procedures and documentation in place, such as spreadsheet operation, backup and restore and training records does not change.
- Your need to comply with 21 CFR Part 11 by having logical security and an audit trail does not change, and therefore your specification would be worded the same for each spreadsheet.

Clearly the evidence that each of these requirements has been met would need to be verified individually at the qualification stage, but at the specification stage many of these needs and responses are identical.

Certain sections of the specification (such as Appendix A, B, and C) will be very different for each spreadsheet. This is the reason these sections are taken out into Appendices; it allows the document writer to know immediately the areas that need close attention in the generation process.

| Cell Address | Cell Format | Cell Type | Formula / Value | Locked | Hidden | Merged | Merged Range |
|--|-------------|------------|--|--------|--------|--------|--------------|
| C10 | 0.00 | Data Entry | | False | False | No | |
| F10 | 0.000 | Data Entry | | False | False | No | |
| G10 | 0.00 | Formula | =IF(ISERROR(AVERAGE(C10:E10)),"",AVERAGE(C10:E10)) | True | False | No | |
| H10 | 0.00 | Formula | =IF(ISERROR(STDEV(C10:E10)),"",STDEV(C10:E10)) | True | False | No | |
| Followed by identical row content until row 999 | | | | | | | |
| C999 | 0.00 | Data Entry | | False | False | No | |
| F999 | 0.000 | Data Entry | | False | False | No | |
| G999 | 0.00 | Formula | =IF(ISERROR(AVERAGE(C999:E999)),"",AVERAGE(C999:E999)) | True | False | No | |
| H999 | 0.00 | Formula | =IF(ISERROR(STDEV(C999:E999)),"",STDEV(C999:E999)) | True | False | No | |

Table. 4 Example of procedure for repeated large spreadsheet

7. Question. If we use macros do we have to printout the source code in the specification?

There is no specific need to list the macro's source code, but with spreadsheet macros it is usually done this way for completeness due to the fact that the code listing is small, and the effort and confidentiality issues over printing are minor.

8. Question. Our spreadsheet is huge, do we have to printout all the formulas?

Normally the whole content of the spreadsheet would be printed out and stored in Appendix B. However, on very large repeated spreadsheets this is often overkill. Many spreadsheets will have the same formulas repeated in an identical fashion for many hundreds or thousands of rows, often the only thing that changes is the row reference in the formula. In this situation we would list the content to shorten the specification content (see Table. 4).

This presents minimal risk, and with recent error checking tools in Excel⁴, if individual rows were inconsistent you would see the error at the prototyping or qualification stage.

9. Question. Our spreadsheet is huge, do we have to printout all the formulas to paper?

Normally this would be done, but sometimes electronic media can be used to help save on paper. Our preferred option would be to run our reporting tools on the spreadsheet and convert the generated table into PDF format and burn it onto CD ROM. This would then stored in a wallet in Appendix B.

10. Question. Our recent inspection/auditor stated that a separate URS was needed for System X? We can't risk combining the URS

or

Our recent auditor stated that source code must be printed for custom programming!

Individual auditors and inspectors will always have their own opinion on many of the points discussed above. The lack of hard and fast rules makes computer systems

validation a difficult area to interpret. In this situation you can only do what you think is correct and defend your approach.

Be careful not to confuse what an auditor asked for with your complex systems with what they will expect with a spreadsheet or standalone instrument, they are not comparable and neither should your validation approach be.

The approach presented in this paper has been through many inspections, and to date it has not been challenged by a regulatory inspector. It has however been challenged on many occasions for not meeting local QA requirements, or for being different to the validation approach taken for other more complex systems.

There is a simple (**but not recommended**) answer to this dilemma, and that is to perform the full validation lifecycle on each individual spreadsheet. Generating a VP, URS, SDS, Source Code Review, FS, IQ, IQR, OQ, OQR, PQ, PQR, VR for each spreadsheet would certainly solve your worries over consistency and completeness. It would however also financially cripple your validation effort and tie up valuable resources in areas which most people would consider unnecessary. You could also ask the question as to whether an extensive 'full lifecycle' suite of documents is any more likely to ensure an error-free spreadsheet? Our belief is that it does nothing to reduce spreadsheet errors, but is far more likely to introduce documentation errors.

The streamlined and pragmatic approach described in this article is about taking measured risks, and presenting the critical information in a clear yet easy to generate format.

Conclusion

A streamlined approach to the specification of Excel spreadsheets has been outlined in this article. This approach condenses best industry practices into one generic specification document. This approach offers a pragmatic way to undertake a cost effective and repeatable process and it can be adapted to simple systems other than spreadsheets. ■

David Howard

BSc CChem MRSC.
Validation Consultant.
ABB Engineering Services
PO Box 99
Billingham, TS23 4YS
United Kingdom
+44 (0)1937 589813 (Office)
+44 (0)7740 051595 (Mobile)
david.howard@gb.abb.com
www.abb.com/lifesciences



Dave Howard is a Validation Consultant at ABB Engineering Services, specialising in End User Computing applications.

David Harrison BSc MBA

Principal Consultant.
ABB Engineering Services
PO Box 99
Billingham, TS23 4YS
United Kingdom
+44 (0)1207 544106 (Office)
+44 (0)7957 635046 (Mobile)
david.harrison@gb.abb.com
www.abb.com/lifesciences



Dave Harrison is a Principal Consultant at ABB Engineering Services where he is the Product Manager for spreadsheet validation solutions.

References:

¹ David A Howard & David Harrison, ABB Engineering Services. "A pragmatic approach to the validation of Excel spreadsheets – Overview". Pharma IT Journal, Vol. 1 No. 2 April 2007.

² DaCS™, Data Compliance System, Compassoft Inc. <http://www.spreadsheetvalidation.com/solutions/dacsproduct.htm>

³ FDA 21 CFR 11, Electronic Records, Electronic Signatures.

⁴ Exchecker™, Compassoft Inc. <http://www.compassoft.com/exchecker.htm>

⁵ PUP, Power Utility Pack, J-Walk & Associates. <http://j-walk.com/ss/pup/pup6/index.htm>

⁶ Good Automated Manufacturing Practice Guide, Version 4, ISPE, Tampa FL, 2001

⁷ R. D. McDowall, "Practical and Cost-Effective Risk Assessment Approaches for the Validation of Commercial Computerised Systems". Pharma IT Journal, Vol. 1 No. 2 April 2007.

⁸ <http://www.spreadsheetvalidation.com>