End User Applications Including Spreadsheets

1 Introduction

This appendix gives guidance on the use of end user applications such as spreadsheets or small databases in a GxP environment.

Application tools are available for creating a wide range of end user applications, including customized statistical analyses, the creation of local databases, data mining, and multivariate analysis. These may be used for GxP regulated activities, and they present particular compliance challenges.

End user applications tend to be among the most under-documented systems used in GxP environments, for the following reasons:

- users regard them as part of the desktop
- the ease with which applications can be built without much training
- the data processing power that they can have

The flexibility and power of the spreadsheet allows users to create tools that range from performing simple calculations to sophisticated analysis of a major clinical study. Special emphasis is placed on spreadsheets in this appendix because PC users may have the opportunity and ability to create a spreadsheet application, and may use them to process regulated data.

The level and rigor of specification and verification applied to end user applications should be based on risk, complexity, and novelty. This appendix provides guidance to help users determine the appropriate approach. While the examples given in this appendix are mainly spreadsheets, the same principles can be applied to other end user applications.

2 Application Types

This section considers typical examples of end user applications found in the GxP environment.

2.1 Disposable spreadsheets

Spreadsheets may be used in the same way as a hand calculator. For example, ten output values from a laboratory test are input for the purpose of calculating a mean and standard deviation. In this scenario, the electronic copy is not retained.

This should be documented in the same way the use of a calculator would be documented, i.e., the values and result are recorded and signed. The results can be printed, labelled, and signed. It should be clear on the page exactly what arithmetic manipulation was done. This can be facilitated in most spreadsheet tools by printing a copy of the spreadsheet displaying the cell formulae. The paper becomes part of the GxP record.
Calculations used to process GxP data should be verified. This does not mean that algorithms used by native functions of the spreadsheet need to be checked for accuracy, but rather to demonstrate that they are the correct calculations. For example, \((a+b)c\) is a very different expression from \(a+(b*c)\), and errors like this are easily made. Verification of the algorithms can be accomplished by using the capability of most spreadsheet software to print the cell formulae, or by a third party review.

### 2.2 Spreadsheets Retained as Documents

In many cases, the way in which spreadsheets are used is more like a word processing document than a traditional application. The main difference is that the spreadsheet can be used to both record GxP data and to manipulate it. The flexibility of manipulation that makes spreadsheets useful makes it advisable to manage them as documents rather than applications. It is likely to be extremely difficult to establish that all subsequent saved copies are the same as the original. Calculations should, therefore, be verified and fully explained, as they would be in a text document. This should include proof that the intended formulae have been used, as described in Section 2.1 of this appendix.

Unless the spreadsheet is adequately controlled, it may be advisable to consider a paper printout as the master record. There are a variety of options for achieving adequate control, including:

- using the spreadsheet tool’s internal security options, such as password protecting cells or sheets
- storing the spreadsheet in a secure directory
- managing the spreadsheet in an electronic document management system

Spreadsheets that are effectively documents should be managed in compliance with the applicable regulations. For example, a common use of spreadsheets is to manipulate and maintain laboratory data, where compliance with electronic record and signature regulations is a particular concern.

### 2.3 Spreadsheets as Databases

Another popular use of spreadsheets is as a simple database, i.e., to manage or store GxP data electronically. Data may be frequently updated, which may cause difficulty because spreadsheets lack the intrinsic controls possessed by many true databases that are necessary to ensure data integrity. For example, spreadsheets generally have limited or no capability to limit a user’s ability to edit data, or to support audit trails where needed. If a compliant solution is to be developed using a spreadsheet, external controls should be developed to overcome these shortcomings. Users should, therefore, be fully aware of the limitations and weaknesses of spreadsheets when proposed as an alternative to a database application.

While there are commercially available products intended to provide audit trail capability to spreadsheets, as a general rule the use of spreadsheets where audit trails are required is inadvisable. Spreadsheet software typically is not designed to provide audit trail functionality and the use of a database with such capability intrinsic to the design is considered preferable.

### 2.4 Template Applications

A very common use of spreadsheets is the development of template solutions, where data can be subjected to a standard manipulation and the result saved as a unique document. Statistical analysis or data mining applications may also fit this subcategory. Templates may be used, e.g., in tabulating and processing data from a clinical study, or similarly, for QC test results prior to product release.
When developing such templates, users and developers should fully understand and document the required manipulation. This allows clear confirmation of design intentions against standard package features to be established and confirmed. The following should also be considered:

- calculations should be verified to be correct
- Will the template be running on a single workstation, or available for download from a single location? If not, how is it ensured that everyone is using the correct version? Version control should be established, supported by an effective change management process.
- How will access to the application and data fields by users and developers be controlled? Ideally, all cells other than data entry should be locked and inaccessible to users.
- How will functionality be configured? Is there a custom script requirement when using application wizards? A macro is custom software. Even when created by keystroke capture, there is a program in a language such as Visual Basic for Applications® (VBA) behind each macro.
- Will there be more than one module? Integration testing is appropriate in such circumstances. For spreadsheets this may involve direct cell links to other worksheets. These links can be affected by changes, and should be addressed as part of the change control process.
- Will data input be only via keyboard? External data feeds need configuration, and a spreadsheet may not be sophisticated enough to deal with unusual input (e.g., a character string that is too long).
- Will output be saved to file or only printed? Electronic record controls may be necessary if the document is retained electronically.

2.5 True Desktop Databases

Both proprietary and open source desktop databases offer superior solutions to managing large volumes of data compared to spreadsheets, but they still are often significantly less secure than more sophisticated database management systems developed to run in IT-managed server-based environments (e.g., Oracle®). This may present significant issues if the information in the database is GxP regulated. External controls may be required.

3 Risk-Based Approach

End user applications can vary significantly in risk and complexity. The following are, however, required for all applications:

- risk assessment and appropriate risk control measures to manage identified risks
- Appropriate specification and verification to determine that the application performs as intended.

The strategy for specification and verification of the application being built should be based on:

- system impact on patient safety, product quality and data integrity (risk assessment)
- system complexity and novelty (architecture and categorization of system components)
- appropriate security to mitigate the risk of unauthorized changes to data or the application
management of the application under change control

Company policies and procedures should define their specific approach to achieving and maintaining compliance and fitness for intended use of end user applications.

This appendix:

- describes how the use of GAMP categories assists with understanding novelty and complexity
- provides advice on appropriate risk-based controls
- provides examples of typical approaches for different applications

### 3.1 Use of GAMP Categories

The product on which the application is built should be considered to be Category 1. Categories for spreadsheets and other end user applications should be viewed as a continuum that spans Categories 3, 4, and 5 (see Figure S3.1). Assignment of a category is a function of the complexity and novelty of the spreadsheet or application. Note, however, that a spreadsheet that merely makes use of the tabular editing power and does no calculations should be considered a document.

A spreadsheet that simply uses native functions to make calculations in place of a hand-calculator is typically Category 3. For example, a laboratory analyst might create a unique spreadsheet to do a calculation related to an “Out Of Specification” investigation. When the spreadsheet’s arithmetic functions are used, the calculations should be fully explained, as they would be in a text document. This should include verification that the intended formulae have been used properly, and that the data being analyzed is the right data. Such verification could easily be documented by having another analyst or a supervisor examine the spreadsheet and approve it. No further verification is required, since there is no need to challenge the accuracy of the calculations.

When developing spreadsheets as templates, such templates could be Category 3 to 5, depending on complexity, see Section 2.4 of this appendix. For example:

- A template is used by analysts in a laboratory to do a routine calculation of averages and standard deviations of experimental results. This is a straightforward arithmetic operation with no configuration, so the template is Category 3.
- A spreadsheet template requires the user to input tablet strength, so that the application automatically branches to different cells to use strength-specific calculations based on this initial input. Such a simple operation would make the sheet Category 4, as it is effectively configured by the analyst before each use of the template.
- a spreadsheet application that employs custom macros or sophisticated or nested logic or lookup functions should be treated as Category 5
Figure S3.1: Continuum of Categories for End User Applications

4 Risk-Based Controls

GxP risk should be assessed. The following aspects should be considered:

- data integrity related to the control of data files, as most end user applications process data
- The complexity of the application, based on the assumption that undetected systemic errors are more likely in software not developed under a rigorous development method, and more complex applications have more opportunities for errors.
- potential impact on patient safety, product quality or data integrity.

Based on these risk assessments, controls should be established which focus on:

- degree of verification
- security control (for both the application code and any GxP records that are in the application)
- control of changes
- control of the infrastructure on which the end user application is built
4.1 Degree of Verification

The extent and rigor of verification should be based on risk, complexity, and novelty.

One level of testing may be appropriate for simple and low risk systems or several levels may be required.

Complex and higher risk applications require more rigorous testing. The amount of logical branching in the application is a good gauge for complexity; if many logic functions (IF, AND, OR, etc.) or lookup tables are used, complexity is higher. Although they are native functions, these introduce more potential pathways through the application, and such branching requires a more sophisticated test strategy.

Macros also increase complexity, because these are effectively embedded secondary applications. Even when created by keystroke capture, there is a program in a language behind it, although macros that simply automate a string of actions are less of a concern than ones that contain logical branches. Macros should be challenged in documented functional testing. Macros that include logical branches should be subject to greater rigor, with attention paid to multiple logic paths.

See Appendix D5 for further details on testing.

4.2 Security Control

Security considerations for end user applications are similar to those for server or web-based applications, such as access to the application, access to data through the application, and access at the operating system level to data or the application code. Security within the environment should be adequate for the type of information stored or processed.

For many end user applications, a combination of infrastructure controls (e.g., restricted access to directories) and controls available through the application (e.g., password protection of spreadsheet cells) can provide some security against unintentional change. These controls may, however, be ineffective in keeping the application author from making changes outside of a change control process, especially if the application resides on an individual workstation. In some cases it may be possible to improve security by running the end user application on a network drive on which the user’s rights are limited, and which includes a regular scheduled back-up process.

Data is often saved within the application itself, especially in spreadsheets. Ensuring adequate data integrity held in spreadsheets requires the use of strict controls, including any required electronic record controls. Where spreadsheets are subject to edit, it is difficult to establish whether original data in subsequently saved copies has been edited. In such cases adequate control can be provided through the use of an Electronic Document Management System (EDMS). Alternatively, it may be necessary to maintain controlled copies in an unalterable format, e.g., PDF or hardcopy. In general, GxP data should not be saved to a non-secure, non-backed up local disk drive.

If the degree of security that can be provided is not adequate for the data being managed, consideration should be given to the use of applications that operate in a more robust environment.

4.3 Change Control

End user applications that process GxP data should be subject to change control. Version management is difficult for such applications. In some cases, especially spreadsheets, management of the application within an EDMS may be an appropriate solution, as an audit trail of application versions will be retained. Another solution is to use library tools that are often used by developers to manage code. These can be used to manage any type of file, can be effective and reasonably easy to implement, and are less expensive than an EDMS. The use of either approach may also control risks related to security of the environment.
As with any change control process, changes to end user applications should have a change record that includes a description of the change and an assessment of the impact. Where appropriate, associated testing should be documented.

### 4.4 Control of the Infrastructure

End user application environments are Software Category 1 (see Appendix M4). These tools provide an application environment for the spreadsheets, databases, programs, or scripts that are developed by users.

The installation of the environment should be verified and the environment should be managed under change and configuration management.

### 5 Examples of Typical Approaches

Figure S3.2 illustrates five different end user applications and a brief summary of potential approaches based on consideration of GxP impact and the complexity of the application. These examples are intended to be illustrative only, and not definitive.

The analysis is based on an assumption of a constant level of risk. If the risk for a particular application is high, then the rigor should be increased.

**Figure S3.2: Examples of Typical Approach Based on Impact and Complexity**

![A Risk-Based Approach to Compliant GxP Computerized Systems](image_url)
A. simple spreadsheet template for arithmetic calculation for content uniformity test:
   • high impact, low complexity
   • recommended approach:
     - User Requirements Specification (URS), documented verification by a third party that the calculations are the right ones
     - security to ensure the sheet is protected against unauthorized change
     - security to ensure the users can access only the approved version
     - secure storage of electronic document

B. spreadsheet record of training attendance
   • low impact, low complexity
   • recommended approach
     - no specific functionality requiring specification and verification.
     - standard controls for electronic documents containing evidence for GxP compliance

C. desktop database for analyzing toxicology study
   • high impact, medium complexity
   • recommended approach
     - full Category 4 approach: validation plan, URS, Functional/Design Specification (may be combined), Traceability, Documented testing against predetermined acceptance criteria, validation report
     - security to limit access to authorized users
     - change control

D. spreadsheet for statistical analysis of a clinical study, with VB macros
   • high impact, high complexity
   • recommended approach
     - full Category 5 approach: validation plan, URS, Functional/Design Specification (may be combined), Traceability, Documented testing against predetermined acceptance criteria, validation report
     - security to limit access to authorized users
     - change control
E. spreadsheet for statistical analysis of manufacturing data for purpose of statistical process control of parameters within validated ranges (includes complex logic and look-up functions).

- low impact, high complexity

  recommended approach:

  - documented verification by a third party that the calculations are the right ones
  - change control
  - security to ensure the sheet is protected against unauthorized change
  - security to ensure the users can access only the approved version

F. desktop database tracking disposition of printed labels

- medium impact, medium complexity

  recommended approach

  - abbreviated Category 4 approach: validation plan, combined URS/Functional/Design specification, documented testing against predetermined acceptance criteria, validation report
  - change control
  - security to limit access to authorized users