

Standard Operating Procedure

Title: Functional Testing Guide for Computerised System

2.2.3.	Alarm Tests	6
2.2.4.	Error Tests	6
2.2.5.	Configuration Tests	6
2.2.6.	Stress Test	6
2.2.7.	Security Tests	6
2.2.8.	Format Tests	7
2.2.9.	Data Integrity Tests	7
2.2.10.	Restart and Recovery Tests	7
2.3.	Extent of Testing	7
2.3.1.	GEP Functional	8
2.3.2.	Minimal Functional	8
2.3.3.	Some Functional	8
2.3.4.	Extensive Functional	8
2.4.	Standard of Testing	8
2.5.	Formality of Testing	9
3.	Guidance on Performing Functional Testing	9
3.1.	Planning	9
3.2.	Review and Approval of the Plan	10
3.3.	Execute the Test	10
3.4.	Evaluate the Test	10
3.5.	Review and Approval of Results	10
4.	Testing to Support Change	11
5.	Appendix 1 - Software Specifications - Useful Criteria	11
6.	Summary of Changes	11

Procedure

1. Introduction / Definition / Purpose of Functional Testing

An overview of the testing philosophy for computerised systems is given in **SOP VAL-040**. Systems are categorised in line with GAMP principles and their functions are listed and rated for GxP impact in Impact Assessment forms, per **SOP VAL-045**. The level of functional testing applicable to each function within the computerised system is rated as “Minimal”, “Some” or “Extensive”. All levels aim to verify that the system functions correctly (performs as specified in normal situations) whereas the increased levels also emphasise verification that the system does not function incorrectly (does not malfunction when challenged with unusual circumstances). The selection of tests also takes the Risk Assessment process into account, per **SOP VAL-055**.

Functional Testing cannot be effective if specifications are either missing or inadequate, or if requirements are written in a manner which is not testable. Refer to [Appendix 1](#) for some examples of software specification requirements.

A range of detailed issues is described below for appropriate inclusion in Functional Test Scripts. Functional Testing can be applied at all levels of software testing, from unit to system level testing. Knowledge of the internal operation of the computerised system (e.g. from a Structural Code Review) and from processes and equipment relevant to the computerised system may be used to enhance Functional Testing and to determine the range of appropriate test conditions and inputs. The extension of Functional Testing may be appropriate where the Structural Code Review yields insufficient information.

Computerised systems are optimally developed to a high standard prior to formal testing. Testing alone cannot fully verify that software is complete and correct. The detection of an undue number of issues on formal testing reduces confidence that the system will operate correctly, despite rectification of the known defects.

In certain circumstances, the manufacturing site may accept either statements of assurance from suppliers based on user experience, or software development standards and the provisions required from the supplier to satisfy site's requirements.

Standard Operating Procedure

Title: Functional Testing Guide for Computerised System

Security testing ensures that the system provides correct access for the specific responsibilities and associated functions according to the predefined security hierarchy. Issues to be verified may include that:

- Different levels of security restrict access as specified, (i.e. that unauthorised access is prevented in accordance with the profile design).
- Passwords are required and meet formatting and updating rules.
- Output data files are unable to be amended either from within the application or by using external programs such as text editors.
- Critical data files are secure from unauthorised deletion or overwriting.
- The ordinary user has access to data entry fields only.
- Updates to program and user-defined configuration files require high-level access.
- Attempts to access system functionality without the correct level of authorisation are logged for review.

2.2.8. Format Tests

Format testing verifies that displayed information and printed reports contain required data, and details of the product/event and of the context of the report. It confirms that the report:

- Identifies its title/purpose.
- Contains descriptive information identifying product/lot and/or GxP event.
- Contains the timestamp of the event being reported.
- Gives the specified data in correct units of measure.
- Contains the date the report was printed and identifies the responsible authority.
- Identifies the application number and software version.

2.2.9. Data Integrity Tests

Data integrity testing focuses on data monitoring and controls. This includes verification of transaction logging to the audit trail, invalid entry checking, error messages, etc. The requirements of **SOP VAL-060** may be used as a guide.

Testing of data migration programs to verify that data will be accurately converted, will enhance confidence in their use. This involves confirmation of data types, indexing of databases and alignment of Tag lists with correct PLC memory addresses.

2.2.10. Restart and Recovery Tests

Testing focuses on specific conditions that may cause the system to terminate unexpectedly. The testing will ensure that the integrity of the data within the application is not compromised in the event of a system, application, system interface, or network failure. It will also ensure that the user can determine the status of operations in process at the time of termination. Testing involves forcing unique conditions (outside of normal application operations) that will result in application termination or significant application errors, e.g.

- Incorrect start-up and shutdown procedures.
- Recovery after power outage without data loss, or loss of cycle information.
- Disaster Recovery Plan for operating programs, user-defined configuration files and data files.

2.3. Extent of Testing

Effective software testing requires that considerable effort be put into the definition of what is to be tested. The appropriate amount of testing to perform depends on the potential impact on Quality and on the risks associated with the system. **SOP VAL-045** is followed to determine the C level (Impact Assessment Category) and consequently the general extent of testing. This will be based on the use of the system, (i.e. its Impact on Quality) and the novelty of the system, (i.e. its GAMP rating). The three GxP test extents of "Extensive", "Some" and "Minimal" (which arise from the C levels of C5, C4 and C3 respectively) are defined in **SOP VAL-045**. GEP Functional testing arises from the C levels of C2 and C1.

Standard Operating Procedure

Title: Functional Testing Guide for Computerised System

3.2. Review and Approval of the Plan

The Functional Test Script is reviewed by the Business/Process Owner (or the Project Manager who is acting on their behalf) to confirm that the proposed testing adequately demonstrates that the computerised system will meet their needs, i.e. complies with Specifications.

It is also reviewed by the Validation Manager or Q.A. Manager for GxP compliance.

The following types of information assist in review of the in the Functional Test Script or protocol:

- Where the report fits into the overall test strategy, e.g. Module Test, Integration Test, SAT, etc.
- Assumptions, e.g.
“Data used is representative of production, both in type and volume”; or,
“The network and technical architecture have remained in a qualified state via Change Control since the last formal qualification”.
- Exclusions, e.g.
“Network and web portals are outside the scope of this testing” or,
“Modules of the program with no impact on GxP and no significant connection to GxP modules are tested under GEP”.
- Limitations, e.g.
“Documented system limitations will not be tested, however workarounds defined for these limitations will be tested”.
- Dependencies, e.g.
“Interfacing equipment and/or instrumentation upon which Functional Testing depends is in a static state and will not change during testing. Any changes made after testing will be considered as possible grounds for re-testing the interfaces to the system”.

3.3. Execute the Test

The environment where test is performed is recorded, e.g. Test, Live. The software version is also recorded, especially where it changes during testing. Where a test is dependant on other tests, the test sequence must be correct.

Test results are recorded with unambiguous “pass” / “fail” statements. Individual results are signed/initialled and dated. Attachment and referencing in the Test Script of screen prints of data values or error messages or prints of graphical displays provides supporting evidence.

Failed results and responses are to be preserved. The Failures Retesting section is filled out where the result is not a pass, and a number is used to track to defects. New Test Conditions and Expected Results are recorded.

The Correction / Failure Cause column is used to:

- Record that a system defect was identified and how it was corrected, (i.e. errors attributable to software or system design),
- Record tester errors and their resolution, (i.e. any mistake that the individual executing the script makes),
- Explain the impact of environment or set-up errors, (i.e. any anomalies that might occur due to the incorrect or incomplete set-up of the test environment or that are caused by environmental incidents such as blackouts).
- Revise the test criteria used validate a function, (i.e. where incorrect test instructions, data or expected results were initially used).

3.4. Evaluate the Test

The conclusion section of the Test Script will contain a summary of the overall test effort, including any discrepancies and their resolution.

A Test Report can be a helpful summary where there are several test scripts.

3.5. Review and Approval of Results

The Functional Test Script is again reviewed by the Business/Process Owner (or the Project Manager who is acting on their behalf) to verify that the computerised system meets their needs, i.e. complies with Specifications.

It is also reviewed by the Validation Manager or Q.A. Manager for GxP compliance.