Goals

When you have completed this unit, you should be able to:

- Identify computer systems with GMP implications within the scope of the GMP facility audit
- Include in the audit an assessment of the computerised systems used to support a GMP facility
- Understand and apply applicable GMP requirements to the audit.
- Recognize compliance or non-compliance of GMP facilities to applicable regulations for computerised systems

This Training Module is written at an introductory level. It aims to equip the GMP auditor to assess the computerised systems used to support a GMP process while performing a general GMP audit.

Where there is likely to be significant focus on the computerised systems, a specialist IS Compliance auditor should accompany the audit team. This may be the case if:

- The GMP facility relies heavily on novel computerised systems
- There are specific IS concerns prior to the audit
- A previous audit has raised concerns about the Facility's computerised systems

Definitions

Supervisory control and data acquisition (SCADA): Application software used for process control and alarm management for the collection of data.

Explanation of Topic

Introduction

Any GMP facility whether involved in contract testing, manufacturing, processing, packaging or distribution will rely to a greater or lesser extent on computerised systems. They are fundamental in ensuring that processes and data are reliable and secure. In order to achieve this they should be maintained in a validated state.

Validating a computerised system is establishing documented evidence that provides a high degree of assurance that it will consistently function in accordance with its pre-determined specifications and quality attributes throughout its lifecycle.

What is an audit of computerized systems at a GMP facility?

An audit of the computerised systems is a review or inspection of the practices, procedures, methods and standards of the GMP facility that are applied during the life-cycle of the computerised system.

validation plan)

Requirements should be unique and prioritized. They should be written at a detailed level, precisely identifying acceptable criteria for success from a users' perspective.

Functional Specification

The Functional Specification (may be referred to as Design Specification) describes how the system is designed to achieve the Functional Requirements. This deliverable is a technical document that identifies the technical solution for the application, underlying software and the hardware necessary to support the system. There should be clear traceability between the requirements and the functional design. This may be achieved using a matrix, cross-referencing, common numbering or any other approach which makes it clear how each requirement is satisfied by the design. It is not necessary to include code or pseudo-code. Some development methodologies may require a formal approach e.g. UML Use Cases. Depending on the complexity of the system it may be appropriate to have a hierarchy of Functional or Design Specification with a high level design specification complemented by more detailed module specific design specifications.

System Programming.

Coding should comply with programming standards for screens, menus, code annotations, etc. Where possible industry standard coding standards should be used. Compliance with coding standards should be assessed through formal code review. A risk based approach can be applied to code review with code selected based on complexity, criticality or experience of the developer.

Defects found during code review should be logged with corrective and preventive actions as appropriate.

Procedures for maintaining and controlling multiple source code versions should be in place. In practice this is often achieved using commercial configuration management software e.g. CVS, Microsoft Visual SourceSafe or IBM Rational ClearQuest.

Where possible separate environments should be used for development and testing. These should be as similar as possible to the production environment for the system in order to assure the validity of system testing. The Development environment is used for developing and unit testing the software. The Test environment is used to test completed components and perform functional and integration testing. The final Production environment is where the application is placed for production roll out.

Test specification

The Test Specification describes the tests to be performed to ensure the system meets the user requirements. Test scripts should be written such that they can be re-executed and the same results can be obtained. Tests should test limit, failure and stress conditions as well as the successful execution of the required functionality. There should be traceability from the requirements, through the Functional Specification to the Test Specification. It should be possible to demonstrate that all required functionality has been adequately tested.

It is good practice for tests to be written by someone other than the developer and executed by another independent person.

Business continuity (BC) and disaster recovery (DR) are two separate, parallel processes are planned for prior to, and executed after, a disaster. BC planning describes how business processes and activities are to continue in the absence of the supporting computerised systems. BC planning depends entirely on the criticality of the system. For non-critical systems the BC plan may be to do nothing until the system is restored.

Disaster Recovery (DR) planning describes how systems are restored to an operational state (including data) and must be in place for all computerised systems, infrastructure and computer centers. Again a risk-based approach is often taken and the DR planning for non-critical systems may be on a ‘best efforts’ basis.

To ensure the plans are effective in the event of a disaster they must be accurate, timely and complete. They must be reviewed and updated periodically and also when prompted by events, e.g. significant reorganizations occur, a disaster or ‘near miss’ incident or the emergence of new risks or threats. Where practical the review should include a test or exercise of the plans.

Security Measures

Security measures should be in place to prevent unauthorized access to the system. Physical security may involve control of access to data center or restricting system access to specific areas within the facility e.g the relevant laboratories or plant areas. Logical access may include controls on accessing the system software, restricting critical system functions to specified users and changing default passwords on operating systems, databases and application software. An important security consideration is keeping the system components updated with security patches provided by the vendor. The aim of both physical and logical security is to protect the system’s functionality and data from unauthorized change.

Electronic Records and Electronic Signatures

With computerised systems supporting the critical GMP processes at a facility and much of the raw data held electronically, it is important that the records and associated signatures are secure, reliable and attributable. It is important that the Facility’s employees understand that their electronic signatures are the equivalent of the hand written signature. Audit trails should record all relevant detail including who created or modified a record and when.

Advanced Topics

Advanced topics are presented for information only. It is unlikely that either of these topics will need to be covered during a routine GMP audit of a supplier. If a ‘for cause’ audit is required and needs to cover these areas in depth it is recommended that an IS Compliance specialist is included in the audit team.

System Retirement and Decommissioning

Method by which a system is phased-out or retired from production service. During this effort the code is archived, data is converted/migrated (if applicable) and the system is dis-assembled and placed out of service. The approach used for system retirement should maintain integrity of data or the necessity of the use of the data in the future.

    - release certificate or note describing the release content and any exceptions.
    - user manuals
    - training material

- Confirm that all master data was in place and verified as correct at time of release.
- Establish if a long-term archival strategy is documented for retaining versions of the software and its associated documentation.
    - Determine what is archived, how long is it retained, where it is archived and how is access restricted.

System Management

- Confirm a detailed system description exists and is maintained.
- Establish how system support is provided.
    - Confirm that the roles and responsibilities of helpdesk, super users and technical support are clear and defined.
    - Ensure they have the training and experience to fulfill their role
- Verify that the system is owned and managed by someone with sufficient authority and budget to maintain it.

Problem Handling

- Confirm that there is a problem reporting mechanism in place that logs reports, tracks their investigation and resolution.
- Verify that both corrective and preventative actions are implemented for faults.
- Establish the link between incident reporting and change management for the system.

Change Management

- Ensure procedures for initiating, authorizing and documenting system changes are in place.
    - Confirm changes are assessed to ensure that the impact on related functionality is understood.
    - Ensure that the management of change includes updates to system documentation, procedural instruction, training and master data as well as the technical functionality.
- Ensure that testing done in support of system changes confirms that related functions are not adversely affected as well as ensuring that the required change is achieved.

Access & Security

- Establish what physical security measures are defined for the system and if they are in effect.
    - Establish how access to physical hardware is controlled.
- Establish what software security measures are defined for the system.
    - Confirm whether access to critical system functionality has been restricted to appropriate users or groups of users.
    - Confirm whether the ability to amend master data has been restricted

- Determine the process by which users have access to the system granted and removed. Determine that the process is current and followed.
    - Confirm that user access is reviewed periodically to ensure users only have the access they require.