REMOVING THE FRUSTRATION from Functional Risk Management

By Chris Reid, Sion Wyn, Charlie Wakeham, and Heather Watson

ISPE GAMP® 5: A Risk Based Approach to Compliant GxP Computerized Systems (Second Edition) [1], section 2.1.4 states "Quality Risk Management (QRM) is a systematic process for the identification, assessment, control, communication, mitigation, and review of risks."

ISPE GAMP® 5, Second Edition [1] states that, "Controls are developed to reduce risks to an acceptable level. Implemented controls are monitored during operation to ensure ongoing effectiveness."

Risks may be managed by elimination by design, reduction to an acceptable level, and verification to demonstrate that risks are managed to an acceptable level. Examples of risk control measures include designing failure detection mechanisms, implementing procedural monitoring, and testing to demonstrate that controls are fit for intended use.

The International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human Use (ICH) harmonized guideline "Quality Risk Management Q9 (R1) [2]" defines two primary principles of QRM:

- "The evaluation of the risk to quality should be based on scientific knowledge and ultimately link to the protection of the patient.
- The level of effort, formality, and documentation of the quality risk-management process should be commensurate with the level of risk."

We can derive from this that effective risk management is dependent on product and process understanding, critical thinking, and an understanding of how a computerized system supports the process.

PLANNING RISK ASSESSMENTS

QRM focuses effort where there is greatest risk to patient safety, product quality, and data integrity. However, experience shows that, too often, risk assessments are purely formal documentation exercises that have limited influence on the computerized system design or the implementation of controls to manage risk. There are a number of considerations when planning the risk assessment approach.

The value of risk assessment is limited by, for example:

- Failure to
  - include subject matter experts with product and process understanding who can clearly define the severity of a process or functional failure or who have the ability to detect a failure.
  - include subject matter experts with the appropriate system or technical understanding who can assess technical complexity and the probability of process or functional failure.
  - define appropriate mitigating actions to reduce risk.
- Risk assessments that are
  - conducted too early, so information is not mature enough to make valued judgement about risk.
  - conducted too late, so there is reluctance to modify design or conduct additional testing to reduce risk.
  - conducted against off-the-shelf functionality that implement industry best practice processes.
  - too granular (e.g., every single requirement is assessed rather than a feature area).

Scaling

High-level risk assessments conducted against processes or application modules can determine whether there is any patient safety, product quality, or data integrity risk. For example, this would include determining whether a finance module has any GxP risk. Processes and/or application modules that have no GxP impact are excluded from more granular functional risk assessments.

Lack of subject matter expertise

Even with the best-designed risk assessment process, a lack of subject matter expertise will lead to an erroneous outcome.

Process owners have process and product understanding and are best placed to determine the severity of risk scenarios. Functional experts have technical understanding of how the functionality is implemented and are therefore best placed to determine the probability of failure.

Generic risks

A number of risks can be anticipated in advance and managed as generic risks, particularly the failure modes. As such, it is not necessary to repeatedly evaluate generic risks. Table 1 provides some examples.

Granular or grouped requirements

Often, every single requirement is evaluated by the risk assessment. When requirements are written at a granular level, it is better to take a holistic approach and assess a related group of requirements, processes, or feature areas. In such cases, the primary risks requiring control can be identified rather than determining unnecessary controls for each specific requirement.

ISPE GAMP® Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management [3] section 2.3.1 stresses the importance of logically grouping requirements in a structured manner. This approach significantly helps with the assessment of related requirements as a feature area or process area. (Note: it may be beneficial from a business perspective to conduct more granular risk assessments.)

Software/configuration error

Risk assessments often record "software/configuration error" as a risk scenario. This is the cause of the failure and is a superfluous and obvious statement. The failure scenario should clearly state how patient safety, product quality, or data integrity could be impacted.

Off-the-shelf solutions

Off-the-shelf solutions are designed by vendors to reflect industry best practice. Solution evaluation should confirm that this is the case. The likelihood of failure is low for mature off-the-shelf solutions. Therefore, for such solutions, the primary focus should be to determine functional severity to define the scope of validation effort. Assessment of probability of failure and detection mechanisms will be of limited value.

Novel solutions

Novel solutions are typically new or innovative solutions that may include limited process or functional understanding. It is essential that sufficient understanding is established before the risk assessment is started. As discussed previously, a risk assessment may evolve iteratively with the solution design. Process owners may require training to better understand the solution prior to commencing risk assessments.

INTEGRATING RISK ASSESSMENTS INTO THE DESIGN PHASE

Risks may be initially evaluated against business processes and/or requirements. However, for complex systems, risks should be further evaluated during the design and configuration phase. Additional information provided by design and configuration records is useful in confirming risk assessments or in revising to address additional information relating to detectability or the likelihood of failure.

Table 1: Examples of anticipated risks.

Feature	Generic Failure Scenarios
Data Interface	Data transfer does not trigger when required Data transfer does not recover following interface failure Data is incomplete Data is not loaded/mapped correctly
Reports	Report layout is incorrect Data is incomplete/incorrect Data resolution is incorrect/inaccurate Report is not generated when required Report is not saved Report is not approved
Data	Data entered by unauthorized person Data entered out of range Data not saved Data not stored accurately or in wrong data item Data approved by unauthorized person Data accessible by unauthorized person Data not protected against loss

Inform Design

Risk assessments are often conducted as a separate activity from process and functional design. As such, risk assessments are used to assess rather than to inform the design. Risk assessment should be an integral part of the design, where the risk assessment evolves with the design to establish a robust solution, rather than being handled by a separate team that conducts the risk assessment to challenge a completed design.

Agile frameworks cater to this very well when the risk assessment is an integral part of user story development, and when it is built into sprint planning, sprint review, and sprint retrospective—when the current risk conclusions and selected controls are developed and continuously monitored.

Evaluate Controls

Often the focus of risk assessment is to formulate test plans and test scenarios. The primary focus of risk assessment should be to ensure the designed controls are adequate. Testing is then used as one approach to verify that designed controls function correctly.

Consider Different Designs

For complex solutions, a different design or configuration may be used to deliver the same requirement. For example, this would