MODERN Requirements

Gold Microsoft Partner

Microsoft

Effective Validation with MR4DevOps

René van Opstal
Consultant and Owner, van Opstal Consulting

van Opstal Consulting

WHITE PAPER

The Life Science industry is evolving. Although it's renowned for innovation—new medicines and medical devices are introduced daily—many Life Science organizations remain conservative when it comes to Computerized System Validation (CSV) and related quality processes. While advanced computer systems are increasingly used in pharmaceutical production, validation activities often remain paper-based. In fact, a 2022 report revealed that 86% of companies still rely on paper-based or hybrid validation methods.

Software solutions such as Microsoft Azure DevOps and Modern Requirements4DevOps (MR4DevOps) offer a significant leap toward paperless, content-based validation. However, merely adopting these tools isn't the end goal; rather, the objective is to establish and implement an efficient, modern validation framework.

After decades of outdated, paper-heavy practices, the industry is finally shifting. A recent International Society of Pharmaceutical Engineering (ISPE) survey found that 74% of participants plan to implement a digital validation toolkit (DVT) by 2024.

This whitepaper provides guidance on defining a paperless, content-centric validation process for Computerized System Validation. Grounded in GAMP®5 2nd Edition and Computer Software Assurance (CSA), this approach is highly risk-based and reflects the industry's move toward more streamlined and effective practices.

Table of Contents

1. Strategy

To establish a paperless validation strategy, it is crucial to work in close collaboration with your Quality Assurance (QA) department. Existing validation procedures may rely on paper-based deliverables and rigid templates, so successfully implementing a paperless approach often requires stakeholder buy-in and updates to current processes. Ultimately, the QA department must agree to the concept before you can apply the new strategy.

Another important factor is **simplicity.** Although automation can streamline your validation workflow, there is a tendency to add complexity by catering to all possible exceptions—many of which occur only rarely. Focus on automating the **core** process steps and avoid overcomplicating the system with edge cases.

When we began developing this strategy, we started with a paper-based, waterfall-style validation approach, as outlined in GAMP®. Figure 1 illustrates a paper-based validation strategy for a Configured system (GAMP® Software Category 4).

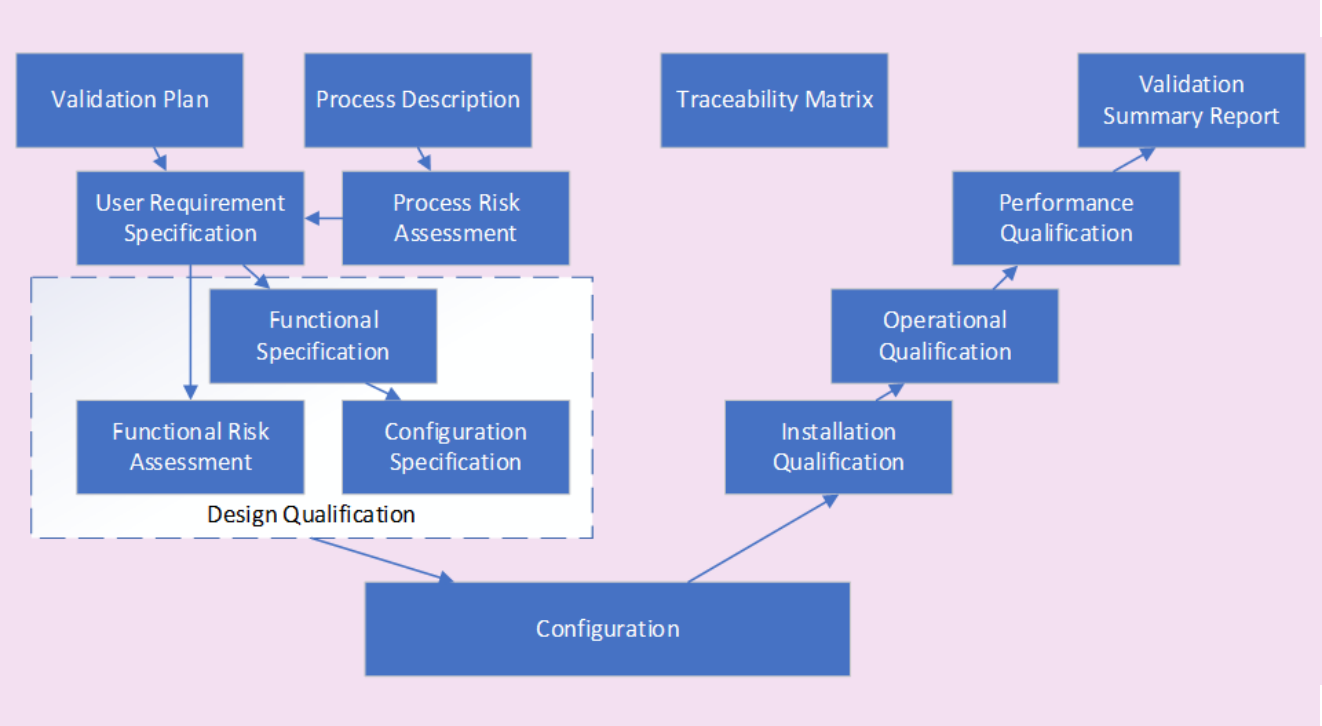


Figure 1 Validation Strategy – Paper-based

By using automated tools, the traditional waterfall approach remains possible, yet adopting Agile techniques becomes significantly easier than in a paper-based system. The most critical change when moving to a paperless validation strategy is adopting a **data-focused** mindset. While automated tools can still generate conventional documents (like the User Requirements Specification, or URS), the real emphasis should be on **managing requirements and data** rather than on producing static documents.

Initially, you may continue approving documents as before. However, as the process evolves, you can shift toward **approving requirements and their related data,** using generated documents primarily to show how different pieces of information connect. These documents can then be exported as reports for external stakeholders such as inspectors and customers.

This white paper's strategy covers all validation deliverables shown in Figure 1 and can be adapted to suit your company's specific validation requirements. For example, if you need a traceability matrix, Modern Requirements4DevOps (MR4DevOps) can automatically create two types. The strategy also includes **approval and versioning** of both requirements and documents, all of which can be performed within Azure DevOps using MR4DevOps.

For the **medical device industry,** MR4DevOps provides a 21 CFR Part 11–compliant environment, including electronic signatures. Documents can also be exported and imported for approval via the Document Management System.

2. Template Design

To implement the validation strategy, you'll need a **process template** in Azure DevOps (located under **Organization Settings)**. In this white paper, we use a "Quality" template based on the default CMMI template (see Figure 2). Keep in mind that after selecting the CMMI process template, you won't be able to choose a different one later, so plan accordingly

Keep in mind that after selecting the CMMI process template you will be tied to this template by default. While you can change the template during a project, doing so may affect all existing requirements and documents. If your organization needs a different process template, consider exploring our free process templates.

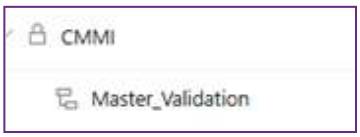


Figure 2: Template selection and creation

Figure 3 shows a high-level overview of the paperless validation strategy used in this Whitepaper.

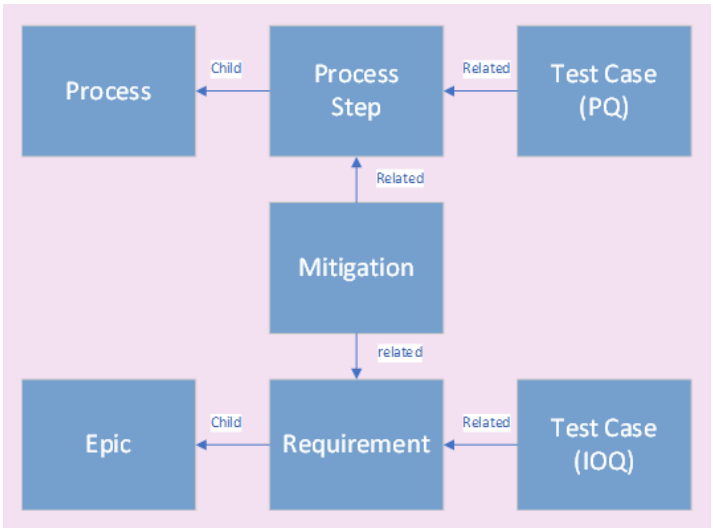


Figure 3: High-level overview Strategy

2.1 Work Item Type

Within the template, Work Item Types are defined. Table 1 (Work Item Types) gives an overview of the most important Work Item Types created or available in the chosen strategy.

Work Item Type	Description	Comment
Process	(Business) process	Created
Process step	Step within process	Created
Epic	Group of requirements	Available

Work Item Type	Description	Comment
Requirement	Requirement	Available
Mitigation	Mitigation action for Risk Assessment	Created
Document	Document used in the strategy	Created for MR4DevOps
Section	Text section in a document	Created for MR4DevOps
Test Plan	Qualification Plan	Available
Test Case	Test Case	Available

Table 1: Work Item types

Other Work Item Types can be created or added as needed but avoid overdoing it. Having too many Work Item Types may result in:

- Increased Complexity: The system becomes harder to navigate and understand.
- Performance Issues: Responsiveness and speed can degrade.
- Maintenance Overhead: Managing and updating numerous Work Item Types requires more effort.
- Inconsistent Usage: Team members may become confused, leading to misuse.

2.2 Process and Process Step

A thorough validation starts with mapping out the relevant processes in which the application is used. The team defines Work Item Types for **Process** and **Process Step,** typically represented in a flowchart alongside a high-level description. In the process template, the team sets up one page for basic information and another page for **Process Risk Assessment (PRA)** data. Tables 2 and 3 show all the fields that the team uses or creates for validation in these **Process** and **Process Step** Work Item Types.

Field	Description	Description	Values
Main Page			
Title	The requirements itself	String	Text
Description	Optional details for the requirement	Rich Text	Text, figures
GMP relevant	Qualification if GMP relevant	Dropdown	Yes, No
Process Owner	Owner for the process	String	Text

Table 2: Process field overview

Field	Description	Type	Values
Main Page			
Title	The requirements itself	String	Text
Description	Optional details for the requirement	Rich Text	Text, figures
GMP relevant	Qualification if GMP relevant	Dropdown	Yes, No
System Impact	Implemented in a system?	String	Text
System	System where step is implemented	Dropdown	Yes, No
Role	Person / role executing the step	String	Text
Risk Assessment page			
Potential Failure	PRA Potential Failure	String	Text
Potential Effects	PRA Potential Effects	String	Text
Severity	FRA Severity	Dropdown	High, Medium, Low
Likelihood	PRA Likelihood	Dropdown	High, Medium, Low
Risk Class	PRA Calculated Risk Class	Dropdown	High, Medium, Low
Current Control	PRA Current Control	String	Text
Detectability	PRA Detectability	Dropdown	High, Medium, Low
Risk Priority	PRA Calculated Risk Priority	Dropdown	High, Medium, Low

Table 3: Process Step field overview

2.3 Requirement

The **Requirement** Work Item Type is central to this strategy. All key information is either contained within or linked to this work item. In the process template, one page is dedicated to basic information, and a second page is created for **Functional Risk Assessment (FRA)** data. **Table 4** lists all the fields that are used or created for validation within the Requirement Work Item Type

Field	Description	Type	Values
Main Page			
Title	The requirement itself	String	Text
Description	Optional details for the requirement	Rich Text	Text, figures
GxP relevant	Qualification whether GxP relevant	Dropdown	Yes, No
Business Priority	Business Priority classification	Dropdown	Mandatory, Optional, Information
Risk Assessment Page			
Potential Failure		String	Text

Field	Description	Description	Values
Potential Effects	FRA Potential Effects	String	Text
Severity	FRA Severity	Dropdown	High, Medium, Low
Likelihood	FRA Likelihood	Dropdown	High, Medium, Low
RiskClass	FRA Calculated Risk Class	Dropdown	High, Medium, Low
Current Control	FRA Current Control	String	Text
Detectability	FRA Detectability	Dropdown	High, Medium, Low
Risk Priority	FRA Calculated Risk Priority	Dropdown	High, Medium, Low
New Likelihood	FRA Likelihood after mitigation	Dropdown	High, Medium, Low
NewRiskClass	FRA Calculated Risk Class after mitigation	Dropdown	High, Medium, Low
New Detectability	FRA Detectability after mitigation	Dropdown	High, Medium, Low
New Risk Priority	FRA Calculated Risk Priority after mitigation	Dropdown	High, Medium, Low

Table 4: Requirement field overview

2.4 Risk

MatCal is used to automatically determine the values for the Risk Class and Risk Priority fields. You team should perform these calculations according to the standards and guidelines outlined in the Risk Assessment definition tables provided in GAMP®5 (as illustrated in Figure 4).

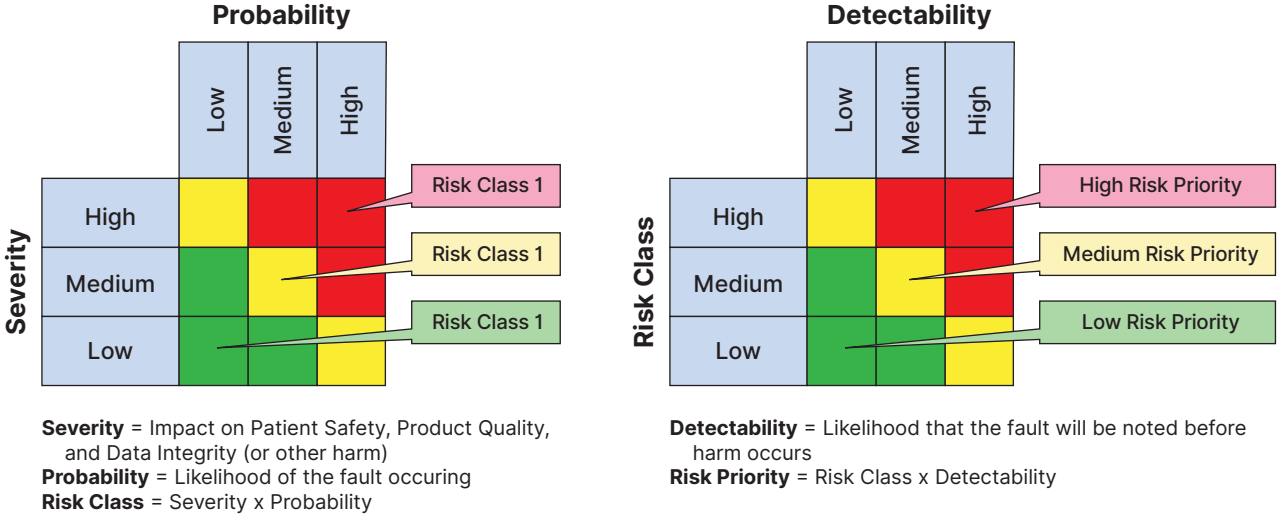


Figure 4: GAMP®5 FMEA Risk definition tables

2.5 State

To manage Work Item Type Requirements, default states should be updated or created. Table 5 gives an overview of the states used. The same states can be used for Work Item types Process and Process Step.

State	Description	Next state
New	New	Defined
Defined	Ready for Risk Assessment	Approved
Approved	Ready for implementation	Implemented
Obsolete	Obsolete	-

Table 5: Requirement field overview

Where the contents of a requirement are modified, the state is reset to Defined. The state Obsolete can be initiated from any other state.

2.6 Mitigation

Your team should include a specific Work Item Type called “Mitigation" to define and track actions aimed at reducing or addressing identified risks. It acts as a structured element in the workflow to document, assign, monitor, and verify the steps taken to manage risks effectively. Table 6 shows the fields used for this Work Item Type, and Table 7 the states.

Field	Description	Type	Values
Title	The mitigation itself	String	Text
Description	Optional details for the mitigation	Rich Text	Text, figures
Mitigation Reference	Reference to the mitigation action	Rich Text	Text, figures
Assigned to	Person to perform the mitigation	Identity	Users
Resolved by	Person that performed the mitigation	Identity	Users
Resolved date	Date Mitigation closed	Date	Date/time
Closed by	Person that Mitigation verified	Identity	Users
Closed date	Date that Mitigation verified	Date	Date/time

Table 6: Mitigation field overview

State	Description	Next state
New	Mitigation created	Accepted
Accepted	Mitigation accepted by Assigned person	Completed
Completed	Mitigation completed by Assigned person	Closed
Closed	Mitigation verified and closed	-

Table 7: Mitigation State overview

3. Approval process

With MR4DevOps, your team can create an efficient and fully digital approval process. This ensures that Azure DevOps maintains a version for all Work Items, but visualizing these versions remains challenging. When the team approves Work Items, they log the approval in the Discussion field of the Work Item, which also appears in SmartDocs as "Rev."

The team can use the review and approval function in MR4DevOps alongside the DevOps state of the Work Items. The team places each Work Item in a specific state (like “Defined”) when all items are ready for approval.

During the review and approval request, the team automatically transfers all approved items into another state like “Approved,” making it clear to stakeholders that the requirements or other document components are approved. Additionally, the team creates Rules in Azure DevOps to make certain fields “Read Only.” They then change the state when they need to modify the requirement.

To simplify the approval process, the team can approve a document. MR4DevOps automatically creates a new version of the document during the approval process. The team can compare new versions of documents with earlier ones.

Users can perform the review and approval of Work Items and Documents without needing a paid license for DevOps or MR4DevOps. The team assigns free stakeholder licenses for this functionality to each user.

4. Requirements Definition

Before creating the requirements, clarify and define the processes that the system supports in a Process Description. Relate many of the requirements to these processes. In this white paper example, the team has not yet implemented the processes in MR4DevOps but plans to do so as a next step to reduce paper usage and improve quality and traceability.

Defining requirements properly is challenging. You need extensive knowledge about different aspects of the system in the scope of validation. GAMP®5 Appendix D1 provides detailed guidance on defining requirements. Consider all the requirement types mentioned in that section. In this whitepaper, the team uses the Epic as the requirement type and marks GxP-critical requirements in the "Requirement - GMP relevant" field.

Consult subject matter experts to account for the wide variety of applicable requirements.
Define the requirements to specify "what," not "how," which the system design explains.

Ensure the requirements follow the S**MART criteria: Specific, Measurable, Achievable, Realistic, and Testable.** Retain ownership of the system and its requirements internally; do not outsource the creation of requirements.

Start building requirements using MindMaps or similar tools. Reference requirements from other systems, as these may already exist in other DevOps projects. Once the requirements are nearly complete, export them into Microsoft Excel and transfer them to DevOps for further refinement. When starting with a new system, create a Project in DevOps and link it to the pre-defined template. Add the requirements to the project during the import from Microsoft Excel.

Use MR4DevOps to create a URS as a SmartDoc. Define several Sections according to the URS document structure in GAMP®5 Appendix D1. Insert the related Requirements into Epics using the drag-and-drop functionality of MR4DevOps.

Use a SmartReport to output a URS in your company's preferred style.

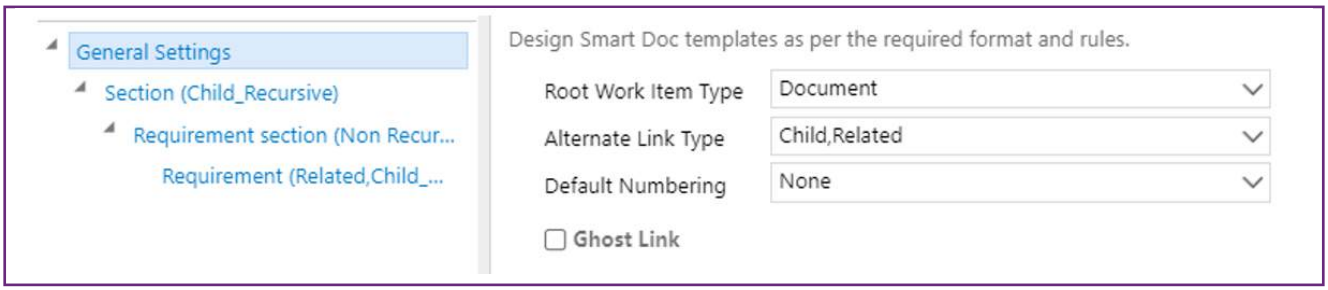


Figure 5: Meta template for URS

5. Specifications

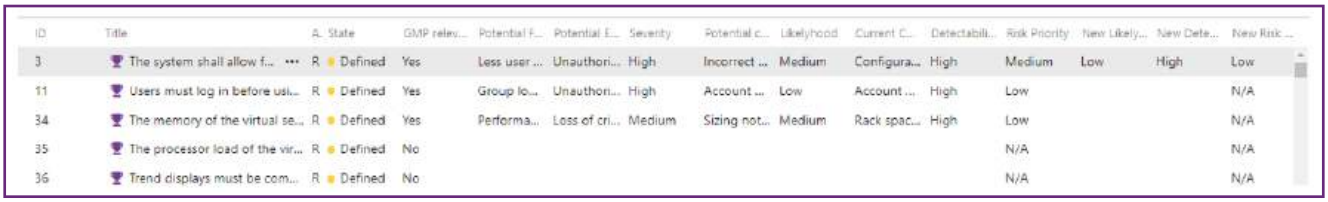
In a paper-based design, Functional Specifications (FS), Software Design Specification (SDS), Hardware Design Specifications (HDS), Configuration Specifications (CS) and other specifications are created. In the current approach of software development, User Stories and Use Cases are used in the design. In MR4DevOps, diagrams can be used to define the User Stories and Use Cases. It is important that these Work Items are always linked to one or more requirements to ensure traceability.

6. Process and Functional Risk Assessment

In a traditional Process and Functional Risk Assessment (PRA and FRA) we assess the risk of each GxP relevant requirement or process step using Microsoft Excel, in a Failure Mode & Effect Analyses (FMEA) using the risk tables shown earlier in this paper. Using MR4DevOps, we take the same approach. The main difference, as opposed to a paper-based approach, is that all information is stored at the requirements. The calculation of the risk is also implemented within MR4DevOps, based on the risk table calculations using MatCal.

For an effective and efficient risk assessment, a proper assessment team is vital. People from multiple disciplines, including QA and a strong assessment leader, must be able to perform the risk assessment in several hours, depending on the number of requirements. It is important to keep the risk assessment simple. High, medium and low risk priorities can be managed, mitigated where required and used to define the test approach.

To perform the risk assessment, create a query in DevOps with all GxP relevant requirements and all required risk related fields of the requirement or process step.



ID	Title	A	State	GMP relev...	Potential F...	Potential E...	Severity	Potential c...	Likelyhood	Current C...	Detectabili...	Risk Priority	New Likely...	New Dete...	New Risk ...
3	The system shall allow f...	R	Defined	Yes	Less user ...	Unauthori...	High	Incorrect ...	Medium	Configura...	High	Medium	Low	High	Low
11	Users must log in before usi...	R	Defined	Yes	Group lo...	Unauthori...	High	Account ...	Low	Account ...	High	Low			N/A
34	The memory of the virtual se...	R	Defined	Yes	Performa...	Loss of cri...	Medium	Sizing not...	Medium	Rack spac...	High	Low			N/A
35	The processor load of the vir...	R	Defined	No								N/A			N/A
36	Trend displays must be com...	R	Defined	No								N/A			N/A

Figure 6: Query example for FRA

Microsoft Excel will still be used as a front-end interface for the FRA. Using the integration features of Microsoft DevOps and Microsoft Excel, the query for the FRA is exported to Microsoft Excel, where the risks are being assessed. Since the calculation of the risk priorities is done in DevOps, this will not be directly visible in the Excel sheet, but the data can be synchronized with DevOps to update the data in DevOps and see the results back in the Excel sheet.

To create an FRA report, and to assign mitigations for medium and High priority risks, again a SmartDoc is used. When the mitigation is defined, the risk priority can be redefined using the Excel sheet again, or directly in the requirement.

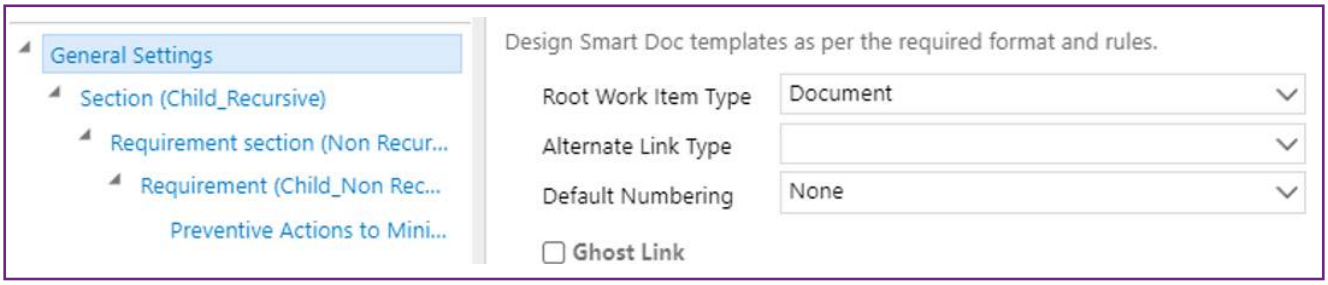


Figure 7: Meta template for FRA report

7. Testing & Qualification

The testing and qualification approach can be used in the same way as in a paper-based validation approach. After module and unit testing by the developers, formal qualification can be done by using Installation Qualification (IQ), or Verification as used in GAMP®5, Operational Qualification (OQ) and Performance Qualification (PQ). The scope and approach for testing must be based on the risk of the requirement defined in the FRA.

In the IQ, the system components and configuration are verified. This will be done by manual testing executed by Subject Matter Experts (SME). For the OQ automatic testing could be used in combination with some manual testing using test scripts or unstructured testing, dependent on the type of requirement and risk.

Manual test cases can be executed in DevOps with a free license. For more sophisticated testing approaches the Basic license for DevOps is required for testers. Additional tools can be used for automated testing. Screenshots and other additional evidence are stored in the test case results. Important is that all test cases need to be linked to the related requirements to create the traceability.

MR4DevOps has several functions to help with creating test cases and creating test protocols and reports. Test Cases can be created using Diagrams; for protocols and reports SmartDocs and Smart Reports can be used.

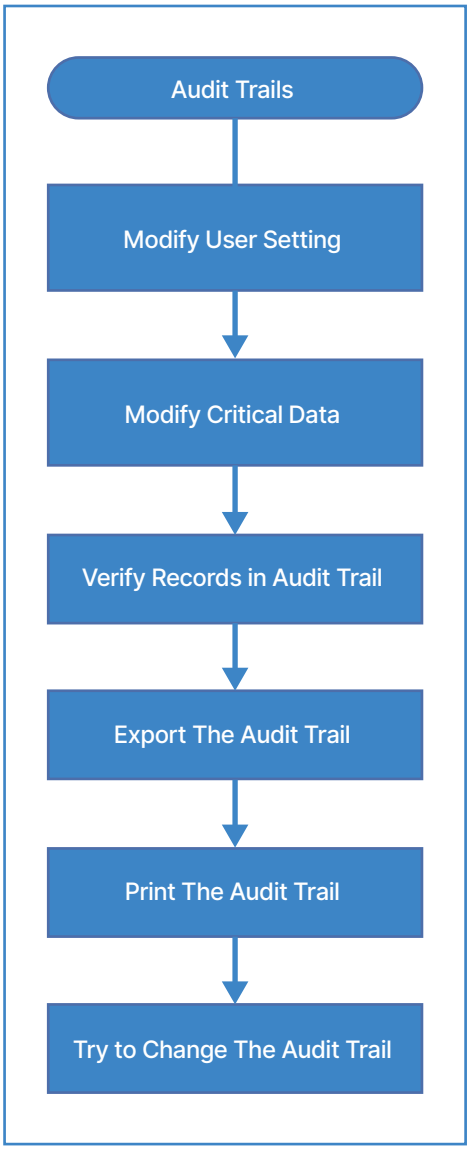


Figure 8: Test case Diagram

8. Traceability

The most valuable and time saving functionality of MR4DevOps is the Traceability function. When all mitigations, design parts and test cases are linked to the related requirements, creating a (horizontal) Traceability Matrix (TM) is just pushing the button after creating a definition with the required columns and information once.

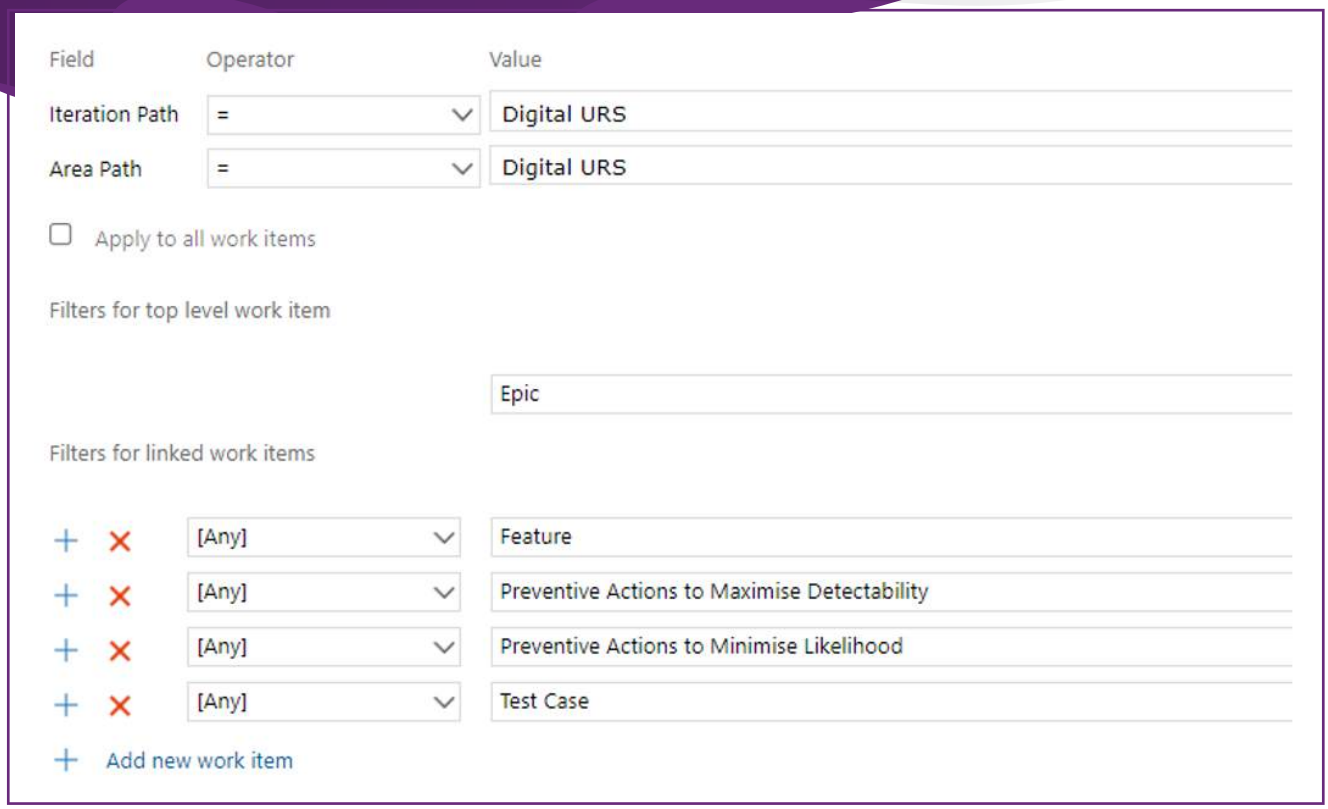


Figure 9: Traceability Definition

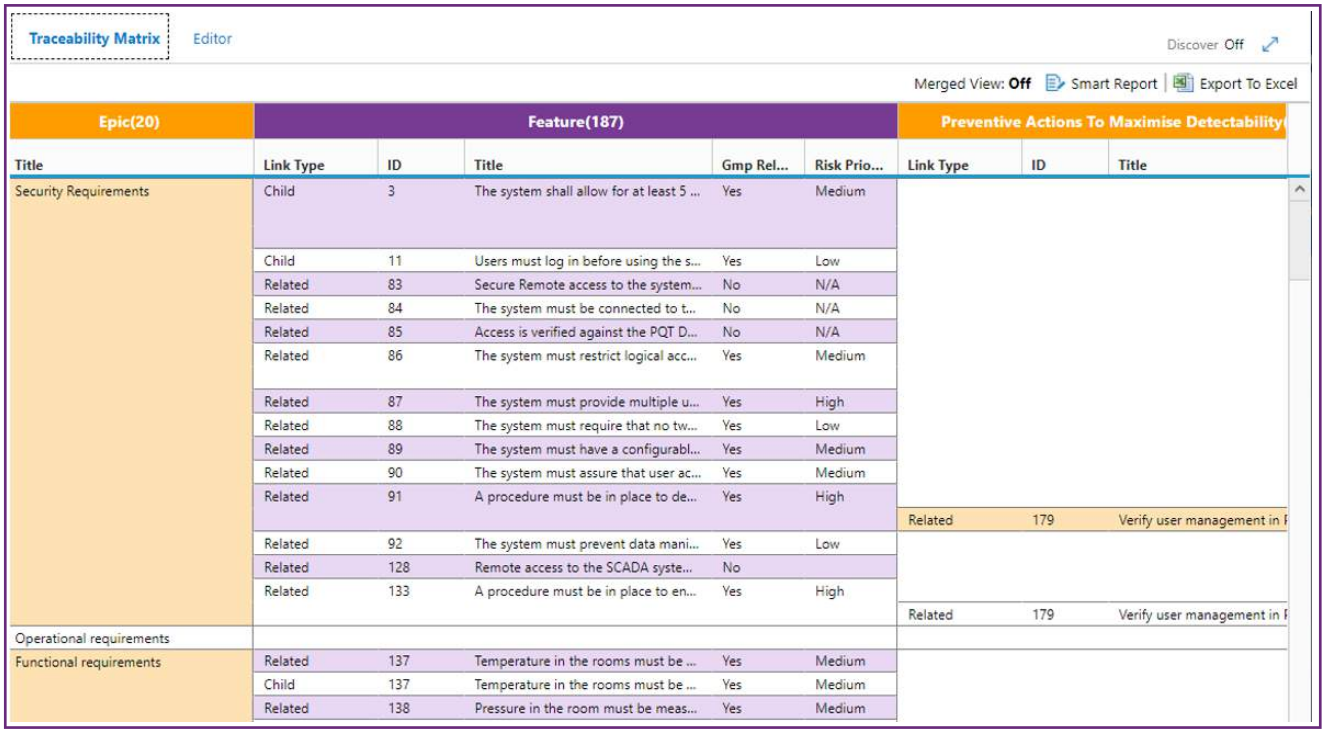


Figure 10: Traceability Matrix

The vertical Traceability function can be used as a management tool to verify whether each requirement has been tested and all mitigations have been performed, and many other verifications you can think of.

9. Procedures and fit for intended use

There is ongoing debate among Computerized System Validation (CSV) experts regarding the level of qualification needed for tools like MR4DevOps. Annex 11 of the European GMP guidelines emphasizes a risk-based approach, suggesting that organizations create an internal procedure outlining how to

implement paperless validation for these types of tools. As always, the QA department should be involved and must approve the chosen approach.

To ensure that MR4DevOps is fit for its intended use:

1. Obtain Vendor Evidence
    - o Confirm that Modern Requirements can provide documentation proving the tool was tested and meets specified standards.
2. Follow Configurations
    - o Set up the tool as described in your internal procedure and as outlined in this white paper.
3. Provide User Training
    - o Ensure that everyone using MR4DevOps is properly trained.
4. Track Versions
    - o For each project, record the version of the tool and manage new releases accordingly.
5. Develop a System Description
    - o Create a simple document describing how the tool is configured.
6. Formally Test Electronic Signature
    - o If required, conduct formal testing of the electronic signature component.

Above all, the QA department needs to be on board with this approach, as their approval ensures compliance and consistency across projects.

Copilot4DevOps

A great feature of MR4DevOps is Copilot4DevOps. This AI feature can make the validation even more efficient and of higher quality. Functions that we often use with the Copilot4DevOps Plus instance in this process are:

- ✓ Generate requirements with acceptance criteria
- ✓ Generate requirements or process step details for design, like User Stories and Use cases
- ✓ Evaluate requirements
- ✓ Generate test cases
- ✓ Evaluate a document against Annex 11 requirements (Copilot Plus)

In conclusion

We hope this white paper provides you with an effective starting point of creating your validation validation approach and environment. Every organization we've supported in validating computerized systems with MR4DevOps has achieved both efficient and effective results.

René van Opstal
Consultant Partner

Optimum Quality
rene.vanopstal@optimumlss.com | www.optimumlss.com

Optimum Quality B.V.
Life Science Support