

Risk-based Approach to Validation

Devora Perich Grieco
Manager, Quality & Compliance

Many organizations using a risk-based approach to validation fail to go through all of the steps, thus increasing time and cost instead of reducing effort.

The Classic and Risk-based Approaches

The FDA defines validation as “establishing documented evidence that provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specifications and quality attributes.” The phrase “high degree of assurance” enables companies to determine for themselves the appropriate level of testing for the system being implemented.

The classic approach to validation involves testing each system requirement in the same comprehensive manner without taking into account how much risk a failure of the function would add to the patient. All functions are comprehensively tested, which may result in testing of positive and negative scenarios to ensure all fault points are equally and exhaustively tested. This approach ensures that every requirement is thoroughly tested, but it can cause unnecessary delays in the release of critical systems, and as such, may actually increase the risk to a patient’s health.

In recent years, companies have begun moving from this classic, all-encompassing validation strategy to a more targeted methodology based on risk. Good Automated Manufacturing Processes (GAMP) 5 seeks to update the industry’s thinking regarding validation resourcing and the appropriate way to allocate resources to get a system released. A functional risk assessment is a way to analyze a system and determine the risk levels of individual functions so that testing can be appropriately scaled to the risk. This contrasts with the classic approach where, by not assessing the risk levels of each function, one is in essence assigning all requirements the same risk level of “High.”

Figure 1 compares the classic approach to validation to the risk-based approach.

Classic Approach	Risk-based Approach
1. Define all requirements.	1. Define all requirements.
2. Choose system to meet all requirements.	2. Choose system to meet all requirements.
3. Perform vendor audit or assessment. Vendor has a business interest in creating a quality system which can be validated, but the responsibility of validating each core function of the system rests with the customer.	3. Perform vendor audit or assessment to gain a high level of confidence in the core software.
4. Design system to meet all requirements.	4. Design system to meet all requirements.
5. Document system core functionality and configuration.	5. Document system configuration, based on function.
	6. Create a function-based risk assessment to rate the risk factor for each function of the system. Trace each configured piece to the appropriate function rating to determine the level of testing required for each.
6. Test every requirement with equal weight.	7. Test every requirement weighted as appropriate.

Figure 1: Process for Classic & Risk-based Approach

It may seem as though the classic approach would take less time, as there are fewer steps. However, in some systems, where the risk level of the company allows it and where there are many requirements that are easily categorized and assessed by function, a significant amount of time can be saved by omitting or limiting formal validation testing on “Low” or “Medium” risk items. Figure 2 below illustrates an example of the differences in time needed for validation efforts when using the classic approach and the risk-based approach.

Task	Days with Classic Approach	Days with Risk-based Approach
User Requirements Specification	15	15
Choose System	15	15
Perform Vendor Audit	15	15
Design System	45	45
Informal Unit Testing	30	30
Functional Requirements Specification	60	60
Functional Risk Assessment	0	7
Testing	203	109
Total	383	296

Figure 2: Example of Validation Effort for Classic & Risk-based Approach

When a little time is taken to perform a risk assessment, a great deal of time can be saved on testing, depending on the risk tolerance of your company. Investing seven days in performing a functional risk assessment can reduce testing time by 46% and decrease project length by 25%, which could translate to an overall reduction of project cost.

The Risk-based Validation Approach

Vendor Audit or Assessment

GAMP 5 recognizes that most off-the-shelf configurable solutions implemented in the quality environment are developed by vendors who have a strong system design lifecycle (SDLC) and internal quality systems to ensure that their products function as designed. This assumption cannot be made without verification, but if a vendor assessment or audit is conducted to ensure a high degree of quality in the software, additional testing may be redundant, depending on the risk of the item. Some vendors of commercial off-the-shelf configurable software (COTSCSW) release and sell their own validation package, which demonstrates that the core functionality of the software is fully validated. If the vendor has demonstrated that it has a quality system in place and has verified (to the level required by the customer) that when the software is configured with X, then Y will happen, then only Y needs to be tested. Additional validation of the core features of the software once the validation package has been formally accepted would be redundant.

User Requirements Specification & Functional Requirements Specification

A process-based user requirements specification (URS) should be created to facilitate traceability and to ensure that a system requirements can be broken into its basic functions when assessing risks. This should not be software specific, but should focus on how the user will utilize the system. Once a system is chosen, the functional requirements specification (FRS) should follow the URS in an easily traceable way. While the FRS does not need to include every aspect of the system, it should indicate the way that the software, once configured, will allow the process to meet the requirements outlined in the URS. In a risk-based approach, all requirements are analyzed for fault points in the system. As it requires a strong understanding of the functions of the system, this should be done by a subject matter expert (SME) in the system being implemented.

Risk Assessment Strategy

A good risk assessment strategy should be agreed upon before the functional assessment begins. The company should take into account any documentation that the software vendor provides to its customers that can lead to assurances that the software itself works the way that it is intended. If the company can assert that the software itself does not need additional qualification (either via the software vendor's own validation documentation or the results of the vendor audit) it may be able to focus testing only on the configuration, rather than re-testing proven functionality. The company should clearly document (in the validation plan, the risk assessment, or other appropriate document) why the strategy used is the right one for this company, including any vendor documentation used.

The first step is to determine an approach and agree on the factors that will result in a risk assignment for each fault scenario. The goal is to determine the impact, likelihood, and detectability of a failure to establish the overall risk of each scenario. A scenario that has a high impact but low likelihood of occurrence and is easy to detect before harm occurs may have a lower risk than something with a medium impact but high likelihood of occurrence and low detectability.

Next, it is important to define what each risk level means to the validation effort with the goal of mitigating the higher risk items to bring their risk levels down. There are several ways to do this:

- Testing the failure point, thus lowering the likelihood of occurrence
- Procedurally controlling the item, thus lowering the likelihood of occurrence
- Actively monitoring the process, thus raising the likelihood of detection

It can be difficult for a quality group accustomed to the classic “test everything comprehensively” approach to accept the idea of not testing certain requirements, but when one assesses each function using predefined criteria, the reasoning behind the methodology becomes clear. For requirements that describe functions that are unlikely to fail or cause harm if a failure occurs, it is logical to rely on informal testing and not require any excessive formal testing.

One way to assess the risk of each system requirement is to break the requirements down into functions. This requires a team consisting of people who:

- Know the software well enough to break down requirements into functionality
- Know the likelihood of each function failing
- Know the system well enough to understand the results of a failure
- Understand the likelihood that the user would detect an issue before harm occurred

A good risk assessment may be reused with minimal modification if another process is added using the same software. The process may take the following steps:

1. Determine the system function categories and describe them in a risk assessment chart. These categories are determined by reviewing all the functional requirements and assessing how each of the requirements correlates to a similar system function. Examples:

- Date/time stamp on electronic signatures
- Required fields
- Locked fields

2. Describe the risk scenarios associated with each system function category, indicating the potential event and any effects that can occur if the system requirement is not available or fails. Positive and negative scenarios may be built into the events (for example, e-signature not required or e-signature erroneously required) as they may have different effects and consequently different risk levels. Each requirement should be referenced in the risk scenario that correlates to it to clarify the level of testing for that requirement and to allow traceability.

3. Determine the severity of impact for each risk scenario by plotting the patient/regulatory impact against the business impact. In GAMP 5, only the patient regulatory impact is stressed, but it does indicate that the business impact may be used to assist in determining the risk. Of course, this method of valuation recognizes that business considerations should never take precedence over patient safety. However, if a failure poses a low risk to the patient but a high business risk, it may be worth it to the company to mitigate that higher business risk to avoid monetary loss upon failure. In contrast, business risk never lowers the patient/regulatory impact; it can only raise it. The next steps are therefore:

- Determine patient/regulatory impact severity for the risk scenario based upon predetermined criteria, such as:

- High: Failure would severely affect patient safety or product quality.
- Medium: Failure may severely affect or would have a minor impact on patient safety or product quality.
- Low: Failure may have a minor impact on patient safety or product quality.

- Determine business impact severity for the risk scenario based upon the predetermined criteria, such as:

- High: Failure would severely affect business processes.
- Medium: Failure may severely affect or would have a minor impact on business processes.
- Low: Failure may have a minor impact on patient safety or product quality.

- Finalize the severity of impact for each risk scenario by plotting the business impact severity against the patient/regulatory impact severity using the following matrix and guidelines:

		Patient/Regulatory Impact Severity		
		Low	Medium	High
Business Impact Severity	Low	Low	Medium	High
	Medium	Medium	Medium	High
	High	Medium	High	High

Figure 3: Severity of Impact Matrix

- Determine likelihood of occurrence due to configuration error for each risk scenario using predetermined criteria.
- Determine probability of detection for each risk scenario using predetermined criteria.
- Determine the risk class for each risk scenario by plotting the severity of impact against the likelihood of occurrence using the following matrix:

		Likelihood of Occurrence		
		Low	Medium	High
Severity of Impact	High	2	1	1
	Medium	3	2	1
	Low	3	3	2

Figure 4: Risk Class Matrix

- Determine the risk priority level for each risk scenario by plotting its risk class against the probability that the scenario will be detected using the following matrix:

		Probability of Detection		
		High	Medium	Low
Risk Class	1	M	H	H
	2	L	M	H
	3	L	L	M

Figure 5: Risk Priority Level Matrix

- Determine the level of testing for each risk scenario by correlating its risk priority level to a level of testing/verification using predetermined instruction:

Risk Priority Level	Level of Testing
Low	No formal testing of requirements beyond what may be tested as a natural progression of test steps. Functionality will only be directly tested in prototype testing.
Medium	Positive testing of requirements.
High	Comprehensive testing of requirements, including negative and boundary testing in addition to positive testing. Additional procedural controls may be required to improve detectability or reduce the likelihood of occurrence.

Figure 6: Level of Testing for each Risk Scenario

Once the assessment is completed, it is best to look at the results sorted by risk priority level in order to be sure that there is consistency in how risks are assigned to items which may have the same effect. If the effect of two functions failing are the same, the impact should be the same even though the likelihood and probability of detection could differ.

Figure 7 shows a graphical representation of the risk assessment:

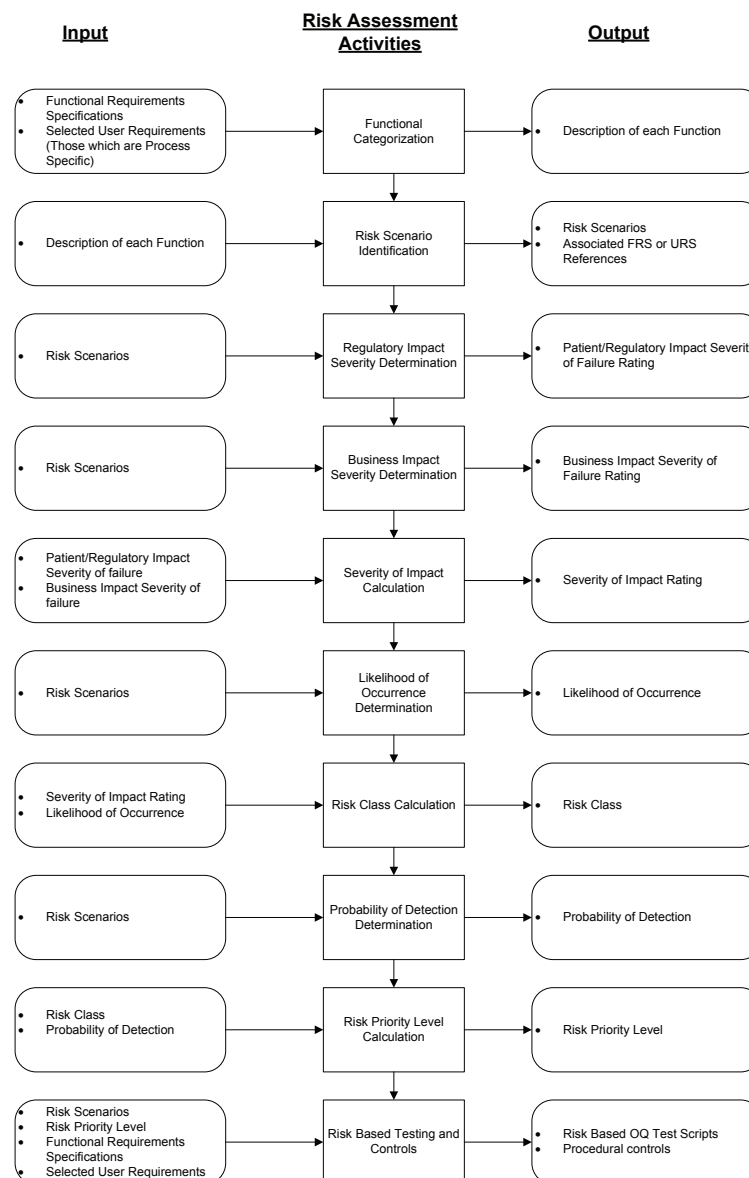


Figure 7: Risk Assessment Process

Risk-based approaches work only if the company implementing a system is ready for the approach. If a firm attempts to use risk-based validation but assigns a high risk to everything or requires the same level of testing for all risks, this will increase time and cost rather than decreasing effort. GAMP 5 specifically discourages just adding the risk assessment to the validation package without truly letting it guide costs in a more practical way.

Let us consider a relatively small system with 1,000 requirements representing 50 functions and 100 events or 2,000 items. With a classic approach, all events must be exhaustively tested for all requirements, leaving 2,000 items to test. With a risk-based approach, the work may be decreased, depending on the risk tolerance of the company. If all risk scenarios are given a high risk level, all requirements will be tested fully, resulting in testing equivalent to the non-risk based approach. In this case, all the company has gained is an unnecessary validation document in the risk assessment. Figure 8 depicts the risk tolerance of the risk-based approach compared to a classical approach by giving each requirement a weight depending on the risk: High (H) =2, Medium (M) = 1, Low (L) =0.

Scenario	% High	% Medium	% Low	Total Weight	Level of Testing	% of Classic Effort
Classic	1	0	0	2	2000	100%
100%H/0%M/0%L	1	0	0	2	2000	100%
0%H/100%M/0%L	0	1	0	1	1000	50%
0%H/0%M/100%L	0	0	0	0	0	0%
50%H/50%M/0%L	0.5	0.5	0	1.5	1500	75%
40%H/40%M/20%L	0.4	0.4	0.2	1.2	1200	60%
33%H/33%M/33%L	0.33	0.33	0.33	0.99	990	50%

Figure 8: Testing Effort of Classic & Risk-based Approaches

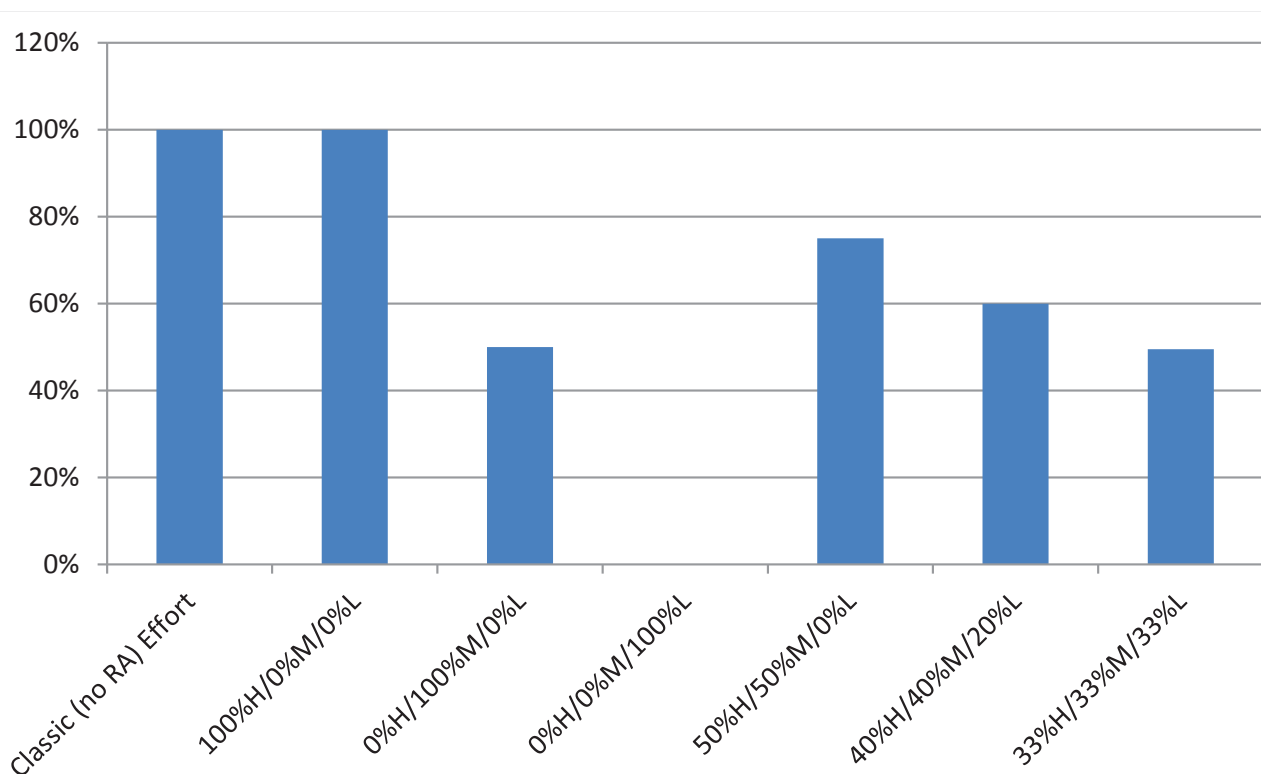


Figure 9: Percentage of Classic Effort

Let us assume that a risk assessment takes seven days to create, the test protocol takes three days to create, and the test scripts take:

- One day per 50 requirements with high level of testing (as they are weighted at 2)
- One day per 100 requirements with medium level of testing (as they are weighted at 1)
- No testing for requirements with low risk weighting

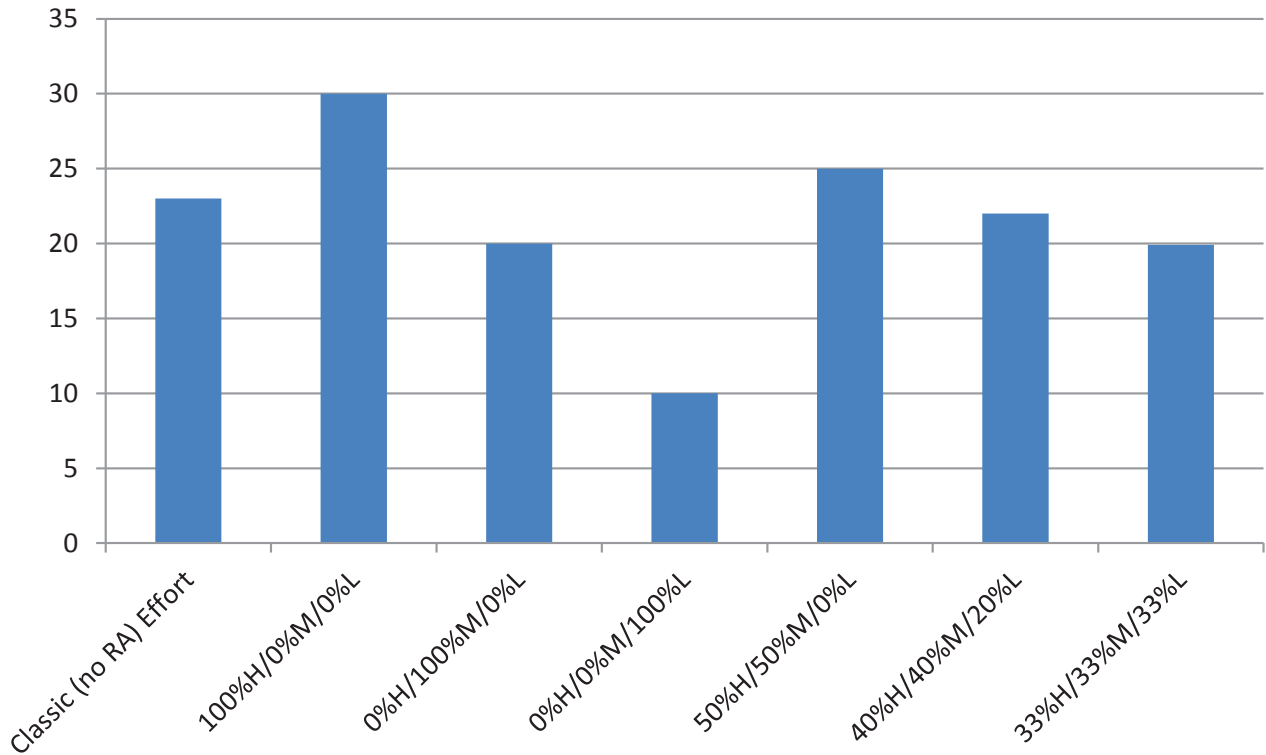


Figure 10: Days of Effort (1,000 requirements)

As Figure 10 indicates, the risk assessment only reduces the days of effort when a company has some level of risk tolerance. This savings increases with larger initiatives, as demonstrated in Figure 11, which considers systems with 3,000 and 10,000 requirements.

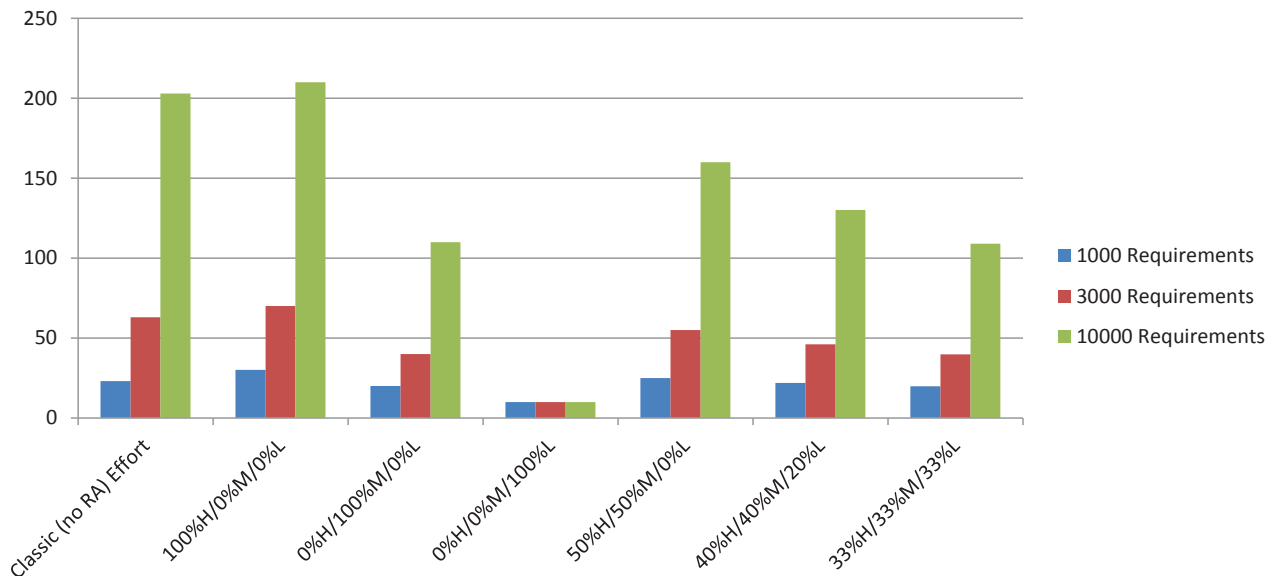


Figure 11: Days of Effort (1,000, 3,000, and 10,000 requirements)

The larger the effort, the greater the overall savings with a functional risk approach. This translates directly to a quicker release to production and a lower project cost.

Conclusion

More and more companies are utilizing risk assessments to guide their validation activities. The functional risk assessment approach may not work for some systems or for companies with a low risk tolerance, but for companies that wish to optimize testing efforts, to release key systems to production in a shorter timeframe, and to get systems up and running sooner and with lower costs, the functional risk assessment can be an invaluable tool.